JMS Journées de méthodologie statistique de l'Insee 2018

Propensity weighting for survey nonresponse through machine learning

Brigitte GELEIN(*), David HAZIZA(**) and David CAUSEUR(***)

(*) Ensai, France ; (**) Département de mathématiques et de statistique, Université de Montréal, Canada ; (***) IRMAR, Applied Mathematic department, Agrocampus-Ouest,France.

bgelein@ensai.fr

Mots-clés. Apprentissage supervisé, machine learning, non-réponse totale, pondération.

Résumé

Recensements, enquêtes ou encore sources administratives, peu importe l'origine des données, elles sont toutes susceptibles de présenter des données manquantes. Le traitement de la non-réponse est d'un intérêt pratique très important étant donnée la baisse constante du taux de réponse aux enquêtes depuis plusieurs décennies. Nous considérons le problème de l'estimation des probabilités de réponse dans un contexte de pondération pour correction de la non réponse totale. Les probabilités de réponse peuvent être estimées par des méthodes paramétriques ou non paramétriques.

La classe des modèles paramétriques inclut la régression logistique comme cas particulier. Les méthodes paramétriques présentent cependant plusieurs inconvénients : (i) elles ne sont pas robustes par rapport à une mauvaise spécification de la forme du modèle, (ii) elles ne sont pas robustes à la non prise en compte d'éventuelles interactions entre prédicteurs ou de termes quadratiques, (iii) elles peuvent conduire à des probabilités estimées très proches de zéro, conduisant à des estimateurs potentiellement instables (Little et Vartivarian, 2005, et Beaumont 2005).

La classe des méthodes non paramétriques comprend notamment la régression par polynômes locaux (Da Silva et Opsomer, 2009), la pondération de classes formées sur la base d'une estimation préliminaire des probabilités de réponse (Little, 1986, Eltinge et Yansaneh, 1997, Haziza et Beaumont, 2007), l'algorithme CHi square Automatic Interaction Detection (CHAID de Kass, 1980), Classification and Regression Trees (CART Breiman et al., 1984, Phipps et Toth, 2012), Conditional inference trees (Ctree) pour des cibles simples ou multiples (Hothorn et al. 2006).

Nous présentons une vaste étude par simulation pour comparer un grand nombre de méthodes d'estimation des probabilités de réponse par apprentissage supervisé, dans un cadre de population finie. Nous couvrons un large champ de méthodes paramétriques ou non, avec des règles de décisions simples ou agrégées telles que Bagging, Random Forests (Breiman, 1996), Boosting (Freund et Shapire, 1996, Friedman et al. 2000) ; voir également Hastie et al. (2009) pour une

revue très complète des méthodes d'apprentissage. Pour chaque méthode, ce sont les performances de l'estimateur par expansion et de l'estimateur de Hajek d'un total qui sont mesurées en termes de biais relatif et d'efficacité relative.

Abstract

We consider the problem of estimating the response probabilities in the context of weighting for unit nonresponse. The response probabilities may be estimated using either parametric or nonparametric methods. In practice, nonparametric methods are usually preferred because, unlike parametric methods, they protect against the misspecification of the nonresponse model.

In this work, we conduct an extensive simulation study to compare methods for estimating the response probabilities in a finite population setting. In our study, we attempted to cover a wide range of (parametric and nonparametric) "simple" methods as well as aggregation methods like Bagging, Random Forests, Boosting. For each method, we assessed the performance of the propensity score estimator and the Hajek estimator in terms of relative bias and relative efficiency.

Introduction

National statistical offices like Insee in France, Statistics Canada or Eurostat at an international level, aim at providing solid foundations for good informed decisions by elected representatives, firms, unions, non-profit organizations, as well as individual citizens. In order to better understand demography, society and economy, analysts and researchers implement statistic methods to analyse data. The latter can be provided by censuses, surveys and administrative sources. Regardless of the type of data, it is virtually certain one will face the problem of missing values. Survey sampling theory meets new fields of research in association with machine learning and big data handling.

Surveys statisticians distinguish unit nonresponse from item nonresponse. The former occurs when no usable information is available on a sample unit, whereas the latter occurs when some variables (but not all) are recorded. Nonresponse may affect the quality of the estimates when the respondents and the nonrespondents exhibit different characteristics with respect to the survey variables. The main effects of nonresponse consist in : (i) bias of point estimators, (ii) increase of the variance of point estimators (due to the fact that the observed sample has a smaller size than the one initially planned), and (iii) bias of the complete data variance estimators (Haziza, 2009). Unit nonresponse is usually handled through weight adjustment procedures (Groves et al. 2001, and Särndal and Lundström 2005), whereas item nonresponse is treated by some form of imputation (Brick and Kalton 1996). These approaches (weight adjustment or imputation) share the same goals : reduce the nonresponse bias and, possibly, control the nonresponse variance.

Turning to weighting adjustment procedures for handling unit non-response, two types of weighting procedures are commonly used (e.g., Särndal, 2007 and Haziza and Lesage, 2016) : in the first, the basic weights are multiplied by the inverse of the estimated response probabilities, whereas the second uses some form of calibration, that includes post-stratification and raking as special cases, for adjusting the basic weights. In this work, we focus on weight adjustment by the inverse of the estimated response probabilities. To protect against a possible model misspecification, it is customary to form weighting classes (also called response homogeneous groups) so that within a class the sample units have similar response probabilities (Little, 1986, Eltinge and Yansaneh, 1997 and Haziza and Beaumont, 2007).

The response probabilities may be estimated using either parametric or nonparametric methods. The class of parametric models includes logistic regression as a special case. There are

several issues associated with the use of a parametric model : (i) they are not robust to the misspecification of the form of the model ; (ii) they are not robust to the non-inclusion of interactions or predictors that account for curvature (e.g., quadratic terms), both of which may not have been detected during model selection ; (iii) they may yield very small estimated response probabilities, resulting in very large nonresponse adjustment factors, ultimately leading to potentially unstable estimates ; e.g., Little and Vartivarian (2005) and Beaumont (2005). In practice, nonparametric methods are usually preferred because, unlike parametric methods, they protect against the misspecification of the nonresponse model. The class of nonparametric methods include kernel regression (Giommi, 1984, Giommi 1987, Da Silva and Opsomer, 2006), local polynomial regression (Da Silva and Opsomer, 2009), weighting classes formed on the basis of preliminary estimated response probabilities (Little, 1986, Eltinge and Yansaneh, 1997, Haziza and Beaumont, 2007), the CHi square Automatic Interaction Detection (CHAID) algorithm (Kass, 1980), Classification and regression trees (Breiman et al., 1984, Phipps and Toth, 2012), Conditional inference trees (Ctree) for simple and multiple targets trees (Hothorn et al. 2006). To estimate the response probabilities in a finite population setting, we cover a wide range of (parametric and nonparametric) "simple" methods as well as aggregation methods like Bagging, Random Forests (Breiman, 1996), Boosting (Freund and Shapire, 1996 and Friedman et al. 2000) ; see also Hastie et al. (2009) for a comprehensive overview of machine learning methods. For each method, we assessed the performance of the propensity score estimator and the Hajek estimator in terms of relative bias and relative efficiency.

1 Theoretical set up

Let $U = \{1, 2, ..., N\}$ be a finite population of size N. In most surveys conducted by statistical agencies, information is collected on a potentially large number of survey variables and the aim is to estimate many population parameters. This type of surveys is often referred to as multipurpose surveys. Let y be a generic survey variable. We are interested in estimating the finite population total, $t_y = \sum_{i \in U} y_i$, of the y-values. We select a sample S, of size n, according to a sampling design $p(S)$ with first-order inclusion probabilities π_i, $i = 1, \cdots, N$. In the absence of nonresponse, a design-unbiased estimator of t_y is the following expansion estimator :

$$\widehat{t}_{y,\pi} = \sum_{i \in s} w_i y_i, \tag{1}$$

where $w_i = 1/\pi_i$ denotes the basic weight attached to unit i.

In the presence of unit nonresponse, the survey variables are recorded for a subset S_r of the original sample S. This subset is often referred to as the set of respondents. Let r_i be a response indicator such that $r_i = 1$ if unit i is a respondent and $r_i = 0$, otherwise. We assume that the true probability of response associated with unit i is related to a certain vector of variables $\mathbf{x}_i$; that is, $p_i = P(r_i = 1 \mid S, \mathbf{x}_i)$. We assume that $0 < p_i \leq 1$ and that the response indicators are mutually independent. The latter assumption is generally not realistic in the context of multistage sampling designs because sample units within the same cluster (e.g., household) may not respond independently of one another ; see Skinner and D'Arrigo (2011) and Kim et al. (2016) for a discussion of estimation procedures accounting for the possible intra-cluster correlation. If the vector $\mathbf{x}_i$ contains fully observed variables only, then the data are said to be Missing At Random (MAR). However, if the vector $\mathbf{x}_i$ includes variables that are subject to missingness, then the data are Not Missing At Random (NMAR) ; see Rubin (1976). In practice, it is not possible to determine whether or not the MAR assumption holds. However, the MAR assumption can be made more plausible by conditioning on fully observed variables that are related to both the probability of response and the survey variables ; e.g., Little and Vartivarian (2005).

If the response probabilities p_i were known, an unbiased estimator of t_y would be the double expansion estimator (Särndal et al., 1992) :

$$\widehat{t}_{y,DE} = \sum_{i \in S_r} \frac{w_i}{p_i} y_i. \tag{2}$$

In practice, the response probabilities p_i are not known and need to be estimated. To that end, a model for the response indicators r_i, called a nonresponse model, is assumed and the estimated probabilities $\widehat{p}_i$ are obtained using the assumed model (e.g., Särndal and Swensson, 1987 ; Ekholm and Laaksonen, 1991). This leads to the Propensity Score Adjusted (PSA) estimator :

$$\widehat{t}_{y,PSA} = \sum_{i \in S_r} \frac{w_i}{\widehat{p}_i} y_i, \tag{3}$$

where $\widehat{p}_i$ is an estimate of p_i. An alternative estimator of t_y is the so-called Hajek estimator :

$$\widehat{t}_{y,HAJ} = \frac{N}{\widehat{N}} \sum_{i \in S_r} \frac{w_i}{\widehat{p}_i} y_i, \tag{4}$$

where $\widehat{N} = \sum_{i \in S_r} \frac{w_i}{\widehat{p}_i}$ is an estimate of the population size N based on the respondents.

The estimated response probabilities in (3) or (4) may be obtained through parametric or nonparametric methods. In the context of parametric estimation, we assume that

$$p_i = f(\mathbf{x}_i, \boldsymbol{\alpha}), \tag{5}$$

for some function $f(\mathbf{x}_i, .)$, where $\boldsymbol{\alpha}$ is a vector of unknown parameters. The estimated response probabilities are given by

$$\widehat{p}_i = f(\mathbf{x}_i, \widehat{\boldsymbol{\alpha}}),$$

where $\widehat{\boldsymbol{\alpha}}$ is a suitable estimator (e.g., maximum likelihood estimator) of $\boldsymbol{\alpha}$. The class of parametric models (5) includes the popular linear logistic regression model as a special case. It is given by

$$p_i = \frac{exp(\mathbf{x}_i^\top \boldsymbol{\alpha})}{1 + exp(1 + \mathbf{x}_i^\top \boldsymbol{\alpha})}.$$

There are several issues associated with the use of a parametric model : (i) they are not robust to the misspecification of the form of $f(\mathbf{x}_i, .)$; (ii) they can fail to account properly on local violations of the parametric assumption such as nonlinearities or interaction effects, both of which may not have been detected during model selection ; (iii) they may yield very small estimated response probabilities, resulting in very large nonresponse adjustment factors $\widehat{p}_i^{-1}$, ultimately leading to potentially unstable estimates ; e.g., Little and Vartivarian (2005) and Beaumont (2005).

In practice, nonparametric methods are usually preferred essentially because, unlike parametric methods, they protect against the misspecification of the nonresponse model. The class of nonparametric methods include kernel regression (Giommi, 1984 and Da Silva and Opsomer, 2006), local polynomial regression (Da Silva and Opsomer, 2009), weighting classes formed on the basis of preliminary estimated response probabilities (Little, 1986 ; Eltinge and Yansaneh, 1997 and Haziza and Beaumont, 2007), the CHi square Automatic Interaction Detection (CHAID) algorithm (Kass, 1980) and regression trees (Phipps and Toth, 2012).

In this work, we conduct an extensive simulation study to compare several methods for estimating the response probabilities in a finite population setting. For each method, we assess the performance of the propensity score estimator (3) and the Hajek estimator (4) in terms of relative bias and relative efficiency. In our study, we attempted to cover a wide range of (parametric and nonparametric) methods ; see Hastie et al. (2009) for a comprehensive overview of machine learning methods.

2 Nonresponse modeling

Estimating the response probabilities is typically a supervised classification issue, in which the response variable is the two-class categorical response indicator r. However, whereas machine learning methods designed to address classification issues usually focus on optimizing prediction performance, we will less ambitiously restrict our attention to the estimation of the posterior class probabilities. For that issue, in some of the statistical learning methods presented below in the present section, it will be considered as a regression issue in which $r = 0, 1$ is treated as a numeric variable.

2.1 Nonparametric Discriminant analysis

Linear logistic regression is often compared to two-class Linear Discriminant Analysis (LDA) since they can both be thought of as different estimations of the same logit-linear regression model, either using maximum-likelihood for linear logistic regression or moment estimation for LDA. LDA originally relies on the assumption that the within-class distributions of the profile $\mathbf{x}$ of explanatory variable is normal with equal variance matrices. Extending LDA to the case of different within-class variance matrices leads to the Quadratic Discriminant Analysis (QDA, see McLachlan 2005). More generally, if $f_r(.)$ stands for the density function of the distribution of $\mathbf{x}$

with class r, for $r = 0,\ 1$, then it is deduced from Bayes' rule that :

$$p_i \quad = \quad \frac{f_1(x_i)P(r_i = 1)}{f(x_i)},$$

where $f(x) = (1-P(r_i = 1))f_0(x)+P(r_i = 1)f_1(x_i)$ is the density function of the two-component mixture model with mixing coefficients $1 - P(r_i = 1)$ and $P(r_i = 1)$.

In a classification perspective, once the within-class distributions are estimated, the predicted class is 1 if the corresponding estimation of p_i exceeds a threshold which is chosen to guarantee a low misclassification rate or a good compromise between true positive and true negative rates. Nonparametric discriminant analysis relies on a nonparametric estimation of group-specific probability densities. Either a kernel method or the k-nearest-neighbor method can be used to generate those nonparametric density estimates. Kernel density estimators were first introduced in the scientific literature for univariate data in the 1950s and 1960s (Rosenblatt 1956, Parzen 1962) and multivariate kernel density estimation appeared in the 1990s (Simonoff 1996). We used a kernel density estimation procedure with normal kernel function, which is the most widely used due to its convenient mathematical properties.

$$\mathcal{K}_r(\mathbf{x_i}) = \frac{1}{(2\pi)^{J/2}d^J|V_r|^{1/2}} exp(-\frac{1}{2d^2}\mathbf{x_i}^\top V_r^{-1}\mathbf{x_i})$$

where J is the number of explanatory variables, d is a fixed radius and V_k the within-group covariance matrix of group r, for $r = 0$ or 1.

2.2 Classification and Regression Tree (CART)

Unlike scoring methods such as logistic regression or discriminant analysis that provide a global decision rule in the range of data, decision trees are designed to search for subgroups of data for which the prediction rule is locally adapted. The CART decision tree (Breiman *et al.*, 1984) achieves this partitioning of the data using a binary recursive algorithm : each split of the learning sample is defined by a binary rule, consisting either in thresholding a quantitative variable or forming two sets of levels of a categorical variable. Decision trees have become very popular in machine learning issues because they can handle both continuous and nominal attributes as targets and predictors.

Once a criterion has been chosen to measure the so-called purity of a group of data, the whole learning dataset, viewed as the root of the decision tree, is optimally split into two children nodes (left and right), so that the sum of the purity indices of the two subgroups is as large as possible. Each of the children node is in turn split following the same goal ... and so on until no further splits are possible due to lack of data. The tree is grown to a maximal size and then pruned back to the root with the method of cost-complexity pruning. Indeed, (Breiman *et al.*, 1984) show that pruning the largest optimal tree produces optimal subtrees of smaller size. Simple or cross-validation assessment of the predictive performance can be used to determine the right size for the decision tree. In order to be able to estimate class probabilities, we choose hereafter to consider $r = 0, 1$ as a numeric variable (which in fact sums up to the use of the Gini index as the impurity measure associated with a unit misclassification cost matrix, see Nakache and Confais, 2003).

Splitting criteria

For each node t which is not terminal, splitting t in two children nodes t_{left} and t_{right} is based on a binary classification rule involving one of the explanatory variables. For each explanatory variable, say x, the binary rule depends on the nature, categorical or numeric, of x. In the case x is nominal, the binary rule just consists in dividing the node t by choosing a group of x levels for t_{left} and the remaining x levels for t_{right}. In the case x is numeric or ordinal, the binary rule consists in a thresholding of x : if the value of x for a given item exceeds a threshold s, then

the item goes to t_{left}, otherwise it goes to t_{right}. The best split is obtained by an exhaustive screening of the variables, and for each variable, by optimization of the binary decision rule. For example, if x is numeric, the optimal choice of the threshold s is achieved by minimizing the sum of within-children nodes sum-of-squared deviations to the mean :

$$\sum_{x_i<s}(r_i - \bar{r}_{t_{\text{left}}})^2 + \sum_{x_i \geq s}(r_i - \bar{r}_{t_{\text{right}}})^2$$

Finally, applying the sequence of recursive binary splitting rules to an item based on its values of the explanatory variables assigns this item to one of the terminal node, say t. The corresponding estimated probability that $r = 1$ is just the proportion $\bar{r}_t$ of respondents in t.

Pruning

Consistently with the above splitting algorithm, if $\mathcal{T}$ stands for the set of terminal nodes of a tree T, then the goodness-of-fit of T can be measured by sum-of-squared differences between the observed and fitted values, namely $C(T) = \sum_{t\in\mathcal{T}}(r_i - \bar{r}_t)^2$. The largest possible tree obtained by applying the recursive binary splitting rules until no further split is possible minimizes $C(T)$. This largest tree may overfit the data, which can be detrimental to its prediction performance. Therefore, it is recommended to prune the tree by minimizing the following goodness-of-fit criterion, penalized by the so-called size $|T|$ of the tree, namely the number of its terminal nodes :

$$C_\alpha(T) = \sum_{t\in\mathcal{T}}(r_i - \bar{r}_t)^2 + \alpha|T|$$

where $\alpha > 0$ is a penalty parameter.

For a given value of α, minimizing $C_\alpha(T)$ results in a unique smallest subtree $T_\alpha \subseteq T_0$. Consistently, progressively elevating α produces a sequence of subtrees $T_0 \supseteq T_1 \supseteq ... \supseteq T_L = t_0$, where t_0 is the complete set of items in the sample. The penalty parameter α is usually obtained by minimization of a cross-validated evaluation of the penalized goodness-of-fit criterion for all the subtrees in the sequence or, as suggested in Breiman *et al.*. (1984) to get more stable results, by taking the subtree which cost is one standard-error above the minimal cost.

Surrogate splits CART handles missing data among regressors with surrogate splits. Breiman proposes to define a measure of similarity between the best split of any node t and any other possible split of t built with a regressor taht is not involved in the best split definition. Surrogate splits are computed by searching for splits leading approximately to the same partition of the observations as the original best split.

In section 3.1, we will see that this way of choosing the optimal tree by pruning is not appropriate for our final purpose of estimating totals on variables of interest that are subject to missingness.

2.3 Conditional Inference Trees for simple and multitarget decision problems

Due to its exhaustive search algorithm for the optimal splitting rules, the above recursive partitioning algorithm has several drawbacks among which overfitting (if not pruned) and selection bias towards covariates with many possible splits. Conditional Inference Trees (Ctree, Hothorn *et al.* 2006) are designed to overcome those two drawbacks by improving the search of the best splitting rules using conditional inference procedures and permutation tests (see Strasser and Weber, 1999).

According to Hothorn *et al.* (2006), conditional inference trees keep the same flexibility as the original tree methods, since they can be applied to different kinds of decision problems, "including nominal, ordinal, numeric, censored **as well as multivariate response variables** and arbitrary measurement scales of the covariates".

Let us assume that, based on a model for the conditional distribution of the response indicator r given a $J-$vector of explanatory variables $\mathbf{x} = (x_1, ..., x_J)^\top$, test statistics can be derived for the significance of the relationship between the response and each of the explanatory variable. As for the standard tree method presented above, the Ctree algorithm to define the optimal splitting rule of a non-terminal node can be divided in two steps :

1. Variable selection : significance of the relationship between the response and each of the explanatory variables is tested, based on random permutations of the response values to obtain a nonparametric estimate of the null distribution of the test statistics. A multiple testing procedure controlling the Family-Wise Error Rate (FWER), such as the Bonferroni correction of the p-values, is then implemented for testing the global null hypothesis H_0 of independence between any of the covariates x_j and the response indicator r. The algorithm is stopped if H_0 cannot be rejected at a pre-specified FWER control level α. Otherwise the covariate x_{j^*} with the strongest association to r is selected.
2. Optimal split : the best split point for x_{j^*} is also chosen using permutation tests for the significance of the difference between the response rates in the two children nodes.

In the above algorithm, the FWER control level α turns out to be the key parameter to determine the size of the final tree.

Predictions

As with CART, in each cell t which is a terminal node, $\hat{p}_i = \bar{r}_t$.

Missing values in regressors

CTree, as well as CART, handles missing data among regressors which is not the case with logistic regression. Surrogates splits are computed by searching for splits leading approximately to the same partition of the observations as the original best split.

2.4 Iterated Multivariate decision trees

Conditional inference trees, introduced in subsection 2.3, can also produce decisions rules with several targets at once (see De'ath G 2002 and 2014). Thus, they enable us to provide **groups of items that can be homogeneous regarding a $Q-$vector of target variables $\mathbf{y} = (y_1, \ldots, y_Q)'$ and the response indicator r.** This could be related with the concept of doubly robustness (Bang and Robins 2005, Haziza and Rao 2006).

In the present item nonresponse context, where all the target variables $y_1, ..., y_Q$ are missing for an item with the target $r = 0$, we propose to implement iteratively MultiVariate CTrees. This procedure can be viewed as an **estimation method of p_i, $i = 1...n$ based on successive steps of simultaneous y imputation.**

1. In the first step, the training sample of the multivariate Ctree is based on the sample of respondents only S_r. The targets are $\mathbf{y}$ and the response indicator r. The predictors are J covariates $x_1, ..., x_J$. In case of missing values among the covariates then surrogate rules can be used. Applying on the nonrespondents sample S_{nr} this first decision tree built on S_r, we get $\hat{\mathbf{y}}$ for non respondents sample S_{nr}.

2. In the second step, the training sample of multivariate Ctree contains all items (respondents and nonrespondents) with observed values of $\mathbf{y}$ for respondents and imputed values (from step one) for nonrespondents. We still use the observed values of the response indicator (not those predicted in step 1) to get new values $\hat{\mathbf{y}}$ for non respondents sample S_{nr}.

3. Step 2 is repeated iteratively until $\hat{\mathbf{y}}$ is stabilized. In our simulation study (section 4), few iterations have been necessary (less than ten). The final output is the n-vector of estimated response probabilities $\hat{p}_i$'s, $i = 1...n$ for each sample item, provided at the last iteration of

multivariate Ctree as the response rate in the terminal node of each item.

This iterated method deals with different patterns of missingness : item nonresponse with imputation of $\mathbf{y}$, unit nonresponse with estimation of response probability and nonresponse among regressors with surrogates rules. It highlights the fact that missingness can be seen as a multivariate problem.

2.5 Bagging and Random Forests

Bootstrap aggregating, also called Bagging "is a method for generating multiple versions of a predictor and using these to get an aggregated predictor. The aggregation averages over the versions when predicting a numerical outcome and does a plurality vote when predicting a class." (Breiman 1996).

This machine learning ensemble meta-algorithm is especially beneficial to the notoriously unstable decision tree methods. It is a special case of the model averaging approach, which aim is both to avoid overfitting and to improve the reproducibility and accuracy of machine learning algorithms.

In a general regression problem, bagging averages predictions over a set of bootstrap samples, thereby reducing the variance of a base estimator (e.g., a decision tree). For each bootstrap sample S_b, $b = 1, 2, ..., B$, drawn in the whole learning sample S_n, a model is fitted with a base estimator, giving prediction $\widehat{f}_b(x)$. The bagging estimate of the response probability p_i, $i \in S_n$ is defined by

$$\hat{p}_i = \widehat{f}_{bag}(\mathbf{x}_i) = \frac{1}{B}\sum_{b=1}^{B} \widehat{f}_b(\mathbf{x}_i)$$

Bagging takes advantage of the independence between base learners fitted on different bootstrap samples to reduce the estimation variance while keeping the bias unchanged. It performs best with strong and complex models (e.g., fully developed decision trees), in contrast with boosting methods (see next subsection) that usually work best with weak models (e.g., small decision trees).

Random Forest (Breiman, 2001) is an extension of Bagging applied to regression and classification tree methods, where the main difference with standard Bagging is the randomized covariate selection. Indeed, to optimize each splitting rule, the Random Forest method first randomly selects a subset of covariates, and then apply the usual split selection procedure within the subset of selected covariates. The former additional randomized feature selection is meant to lead to more independent base learners leading to a more efficient variance reduction, in comparison with Bagging. The Random Forest method usually has a worse starting point (when $b = 1$) than Bagging but converges to lower test errors as B increases (Zhou, 2012).

Note that we have chosen to aggregate within a family of learning algorithm, both in Bagging and Random Forest, and not in an overall perspective mixing different families - unlike in stacking (Wolpert 1992, Breiman 1996, Nocairi et al. 2016).

2.6 Gradient Boosting and Stochastic Gradient Boosting

Similarly as in the Bagging methods, Boosting aims at taking advantage of a set of classification methods, named learners, to improve the overall classification performance. The original learners are assumed to be just slightly better than random guessing : for this reason, we talk about weak learners. The basic principle of Boosting is to iteratively derive a performant classification rule by selecting a weak learner at each iteration and combine it with the learner derived at the preceding step in such a way that the items with largest prediction errors are especially

targeted by the current update of the boosted learner. Boosting was first proposed in the computational learning theory literature (Shapire 1990, Freund 1995, Freund and Shapire 1997) and rapidly became popular since it can result in dramatic improvements in performance.

Friedman *et al.* (2000) give a more statistical perspective to boosting by using the principles of additive modeling and maximum likelihood. Hastie *et al.* (2009) argued that decision trees are ideal base learners for applications of boosting. This motivates our choice of boosting decision trees in our study.

One of the most famous family of boosting methods is Adapative Boosting (AdaBoost, Freund and Shapire, 1996). Hereafter, we present a variant of Adaboost, named Real Adaboost (Freund and Shapire 1996, Schapire and Singer 1999, Friedman et al. 2000), especially suited to the present purpose of estimating response probabilities rather than predicting the membership to the group of respondents. Indeed, at each iteration b, $b = 1, ...B$, the Real AdaBoost algorithm uses weighted class probability estimates $\hat{p}_b(x)$ to build real-value contributions $f_b(x)$ to the final aggregated rule $F(x)$, i.e. to update the additive model. In the following, the base learners $h_\gamma^{(b)} : \ x \mapsto h_\gamma^{(b)}(x) = \pm 1$ (+ 1 for respondents and -1 for non-respondents) are B decision trees with a number γ of terminal nodes.

Real AdaBoost

Input : Learning sample S_n,
Base learning algorithms $h_\gamma^{(b)}$
Number of iterations B,

Process :

1 : Initialize the boosted estimator $F^{(0)}(x) = 0$ and weights $w_i^{(0)} = \frac{1}{n}$, $i \in S_n$

2 : **For** $b = 1$ to B **do**

a : Fit $\hat{h}_\gamma^{(b)}$ with the target $\tilde{r_i}$ (where $\tilde{r_i} = 1$ if $r_i = 1$ and $\tilde{r_i} = -1$ if $r = 0$)
on the weighted items in the training samples, using weights $w_i^{(b)}$,
in order to obtain class probability estimates $\hat{p}_b(x_i)$

c : Update

· $w_i^{(b+1)} = w_i^{(b)} \ exp\{-\tilde{r_i} f_b(x_i)\}$, $i \in S_n$, with $f_b(x_i) = 0.5 \ log\{\frac{\hat{p}_b(x_i)}{1-\hat{p}_b(x_i)}\}$
and renormalize so that $\sum_{i \in S_n} w_i^{(b+1)} = 1$

· $\hat{F}^{(b)}(x) = \hat{F}^{(b+1)}(x) + f_b(x)$

End for

Outputs :

· The classifier $sign[\hat{F}^{(B)}(x)]$ estimates the label

· The estimated probability
$\hat{p}(\tilde{r} = 1|x) = \hat{p}(r = 1|x) = \frac{1}{1+exp(-2\hat{F}^{(B)}(x))}$

In our study, the more sophisticated Gradient Boosting and Stochastic Gradient Boosting versions (Friedman 2002, Culp et al. 2006) of Real AdaBoost are implemented.

Gradient Boosting is a mix of gradient descent optimization and boosting. Both Boosting and Gradient Boosting fit an additive model in a forward stage-wise manner. In each stage, they both introduce a weak learner to compensate the shortcomings of previous weak learners. However, Gradient Boosting especially focuses on the minimization of a loss function, here the exponential loss function derived from the maximum-likelihood estimation of a logistic regression model, by identifying those "shortcomings" using gradients, instead of the AdaBoost weighting function : "Both high-weight data and gradients tells us how to improve the model", (Li 2016). In addition, a regularization parameter is introduced to control at each iteration the weight of the new learners in the current update of the boosted classification method.

The Stochastic Gradient boosting algorithm is referred to as a hybrid bagging and boosting algorithm (Friedman 2002), in the sense that it combines advantages of the two procedures : at each iteration, the new learner is not fitted on the whole learning sample but on a randomly drawn subsample.

2.7 The Suppport vector Machine

Support Vector Machines (SVM) are among the most famous machine learning methods in the statistical learning theory presented in Vapnik (1998). In the special case where the p-dimensional space of data points $(x_{i1}, \ldots, x_{ip})$, where x_{ij} is the observation of the jth explanatory variable on the ith sampling item, is fully separable into two subgroups, one with only respondents and one with only non-respondents, using a linear combination of the explanatory variables, then there exists two parallel hyperplanes separating the two subgoups, with maximal distance between those two hyperplanes : this maximal distance is named an hard margin. The maximal-margins hyperplanes contains data points that are called the support vectors. In this special case of separable groups of respondents and non-respondents, the linear SVM classifier consists of considering the position of a data point with respect to the hyperplane that lies in the middle of the maximal-margins hyperplanes to determine the class of an item.

In the general case where the space of data points $(x_{i1}, \ldots, x_{ip})$ is not fully separable, whatever the hyperplane and the margin chosen to separate the two subgroups, any linear classification rule defined as in the fully separable case by the position with respect to a separating hyperplane will result in misclassified data points. A so-called hinge loss function, very similar to the deviance loss function minimized in the maximum-likelihood estimation of a logistic regression model, is introduced to measure the relevance of a linear classification rule in-between the two maximal-margin hyperplanes. For a given soft margin, finding the optimal hyperplane can be stated as minimizing the mean hinge loss over the learning sample, which is convex optimization issue. The SVM solution finally consists in choosing the best compromise between a low mean hinge loss over the learning sample and a wide margin.

One of the reason why SVM has become so popular is that it can easily be extended to non-linear classification rule, using the so-called "kernel trick" (Schölkopf and Smola 2002). Indeed, in the linear framework, both the mean hinge loss function and the squared inverse of the margin size involve standard scalar products $x_i.x_{i'}$ of data points i and i'. This standard scalar product can be replaced by $K(x_i, x_{i'})$, where K is a symmetric positive definite kernel function (Hastie *et al.*, 2009), that is intentionally introduced to define the similarity of two observations, after a nonlinear transformation of the explanatory variables : to each choice of K corresponds a nonlinear transformation φ such that $K(x_i, x_{i'}) = \varphi(x_i).\varphi(x_{i'})$. For example, the gaussian radial kernel, that is used in the following because it is a "general-purpose kernel used when there is no prior knowledge about the data" (Karatzoglou et al. 2006), is defined as follows :

$$K(\mathbf{x}_i, \mathbf{x}_{i'}) = exp(-\gamma \sum_{j=1}^{J} (x_{ij} - x_{i'j})^2)$$

where γ is a positive constant.

It can be shown that the SVM classifier can be expressed as the sign of a score function $\hat{f}(\mathbf{x})$ which is straightforward deduced from the hinge loss function. Since we are more interested in estimating class probabilities than in predicting class labels, we use Platt's a posteriori probabilities (see Platt, 2000) :

$$\widehat{P}(r = 1|\hat{f}(\mathbf{x})) = \frac{1}{1 + exp(A\hat{f}(\mathbf{x}) + B)}$$

where A and B are estimated by minimizing the negative log-likelihood function.

3 Modifications of "raw" probabilities estimations

3.1 Homogeneous Response Groups (HRG)

The different methods listed above produce "raw" estimated probabilities. The survey weights may be then adjusted inversely to those raw estimated response probabilities. But in order to protect against model insufficiency, it is suggested that homogeneous response groups be formed, i.e. that units with the same characteristics and the same propensity to respond be grouped together (Eltinge and Yansaneh, 1997, Haziza and Beaumont, 2007, Little, 1986). That is why we computed, for each set of "raw" estimated probabilities, a corresponding Homogeneous Response Groups (HRG) version.

Defining HRG requires to partition the population into C groups. The design weight of respondents in group c is adjusted by multiplying it by the inverse of the observed response rate in class c, for $c = 1$ to C. Homogeneous groups are formed by using a clustering algorithm (k-means) on "raw" estimated probabilities. Finally, the probability of a unit in class c is estimated by the response rate observed in the same class.

Example of HRG's usefulness with CART :
CART pruning consists in selecting a tree minimizing a cross-validated error (see section **??**). Therefore, the way the learning method is optimized is not especially designed to match our final aim which is to minimize the following expected estimation error :

$$E(t_y - \widehat{t}_y)^2$$

Therefore, in the following simulation study, we propose to extract clusters of homogeneous estimated response probabilities calculated using unpruned trees.

Let us take as example, the variable of interest Y_1 and response mechanism $R0$ described bellow in the simulation study section 4. With this example, we measure a bias of 11% for the expansion estimation $\hat{t_{y}}_{Exp}$ of t_y in our simulation study with a default pruned CART leading to 6 splits but no bias with an unpruned tree (see figure 1 below). Furthermore, the SSE of $\hat{t_{y}}_{Exp}$ is much lower with 40 splits than with 6 splits.

3.2 Truncation of estimated probabilities

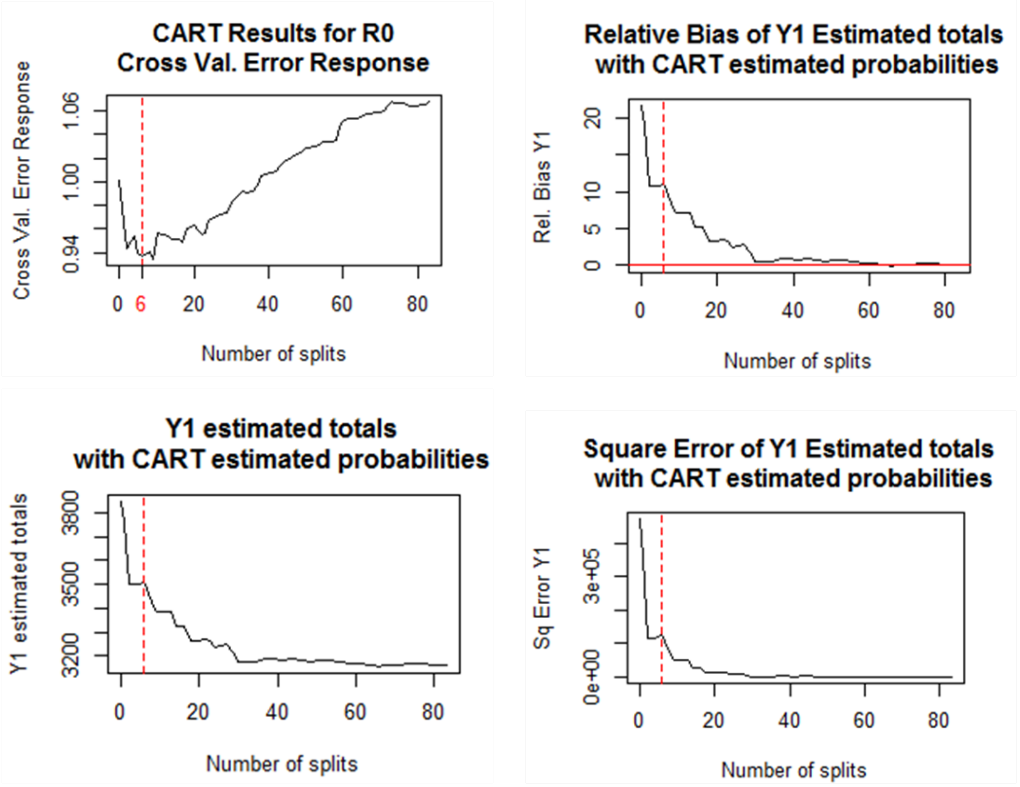
In order to prevent from too small weights, a lower bound has to be determined for the $\hat{p}_i$'s. In practice, the lower bound 0.02 is often used. However, some of our simulations show that the choice of the lower bound may have a certain impact depending on the machine learning method in use. For instance, in our simulations, the global performance of Ctree is robust to variations of the truncation level, which is not the case with the Bagging version of Ctree (see appendix 6.3 for details).

4 Simulations study

4.1 Simulations set-up

We conduct an extensive simulation study to compare the different methods described in Section 2 in terms of bias and efficiency. We perform $K = 1000$ iterations of the following process : first, a finite population of size $N = 1500$ is generated from a given model. Then, from the realized population, we generate nonresponse according to a specific nonresponse mechanism. Below, we describe one iteration in further details.

FIGURE 1 – Performance of CART depending on the number of splits



We generate a finite population of size $N = 1500$ consisting of ten survey variables, y_j, $j = 1, ..., 10$ and five auxiliary variables x_1-x_5. First, the auxiliary variables were generated as follows. The x_1-values are generated from a standard normal distribution. The x_2-values are generated from a beta parameter with shape parameter equal to 3 and scale parameter equal to 1. The x_3-values are generated from a gamma distribution with shape parameter equal to 3 and scale parameter equal to 2. The x_4-values are generated from a Bernoulli distribution with probability equal to 0.7. Finally, the x_5-values are generated from a multinomial distribution with probabilities $(0.4, 0.3, 0.3)$. We standardize x_2 and x_3 so that their means equal zero and their variances equal one : without loss of generality, it allows us to have more readable coefficients in the definition of the models $M1$ to $M10$ and of response mechanisms $R0$ to $R6$ provided bellow.

Given the values of x_1 to x_5, the values of y_1-y_{10} were generated according to the following models :

$$
\begin{aligned}
M1 &: \; y_{i1} = 2 + 2x_{i1} + x_{i2} + 3x_{i3} + \epsilon_{i1} \, ; \\
M2 &: \; y_{i2} = 2 + 2x_{i1} + x_{i2} + 3x_{i3} + \epsilon_{i2} \, ; \\
M3 &: \; y_{i3} = 2 + 2x_{i4} + 1.5 \; \mathbf{1}_{(x_{i5}=1)} - 2 \; \mathbf{1}_{(x_{i5}=2)} + \epsilon_{i3} \, ; \\
M4 &: \; y_{i4} = 2 + 2x_{i1} + x_{i2} + 3x_{i3} + 2x_{i4} + 1.5 \; \mathbf{1}_{(x_{i5}=1)} - 2 \; \mathbf{1}_{(x_{i5}=2)} + \epsilon_{i4} \, ; \\
M5 &: \; y_{i5} = 2 + 2x_{i1} + x_{i2} + 3x_{i3}x_{i4} + 1.5 \; \mathbf{1}_{(x_{i5}=1)} - 2 \; \mathbf{1}_{(x_{i5}=2)} + \epsilon_{i5} \, ; \\
M6 &: \; y_{i6} = 2 + 2x_{i1} + x_{i2}^2 + 3x_{i3} + \epsilon_{i6} \, ; \\
M7 &: \; y_{i7} = 2 + 2x_{i1}^3 + x_{i2}^2 + 3x_{i3}x_{i4} + 1.5 \; \mathbf{1}_{(x_{i5}=1)} - 2 \; \mathbf{1}_{(x_{i5}=2)} + \epsilon_{i7} \, ; \\
M8 &: \; y_{i8} = 1 + exp(2x_{i1} + x_{i2} + 3x_{i3}) + \epsilon_{i8} \, ; \\
M9 &: \; y_{i9} = 1 + x_{i4}exp(2x_{i1} + x_{i2} + 3x_{i3}) + \epsilon_{i9} \, ; \\
M10 &: \; y_{i10} = 1 + 4cos(x_{i1}) + \epsilon_{i10}.
\end{aligned}
$$

As a first step away from our simplest linear model $M1$, for y_2 we only modify the errors : they are generated from a mixture of a standard normal distribution and a beta distribution with

shape parameter equal to 3 and scale parameter equal to 1. For the other variables $y_3, ...y_{10}$, the models are more complicated in terms of relations between variables of interest and covariates but the errors ϵ_{ji} are generated from a standard normal distribution.

In order to focus on the nonresponse error, we consider the case of a census that is, $n = N = 1500$. In each population, response indicators are generated according to the following response mechanisms. The response mechanism $R0$ is a logistic model and constitutes the reference model in our empirical study. The other response mechanisms $R1$- $R5$ are expressed as the sum of p_0 and different terms that draw them away from the reference model. The response mechanism $R6$ is built as a regression tree decision rule.
In each population, seven sets of response indicators r_{id} are generated independently from a Bernoulli distribution with parameter p_{id} (i.e. response probabilities), $i = 1, \cdots, N$ and $d = 0, \cdots, 6$, which leads to seven sets of respondents.

$$
\begin{aligned}
R0 :\ & p_{i0} = 1/[1 + exp\{-0.4(6.5 + 2x_{i1} + 2x_{i2} + 2x_{i3} - x_{i4} + 1.5\ \mathbf{1}_{(x_{i5}=1)} \\
& - 2\ \mathbf{1}_{(x_{i5}=2)} - x_{i3}x_{i4})\}]\,; \\
R1 :\ & p_{i1} = 0.65p_{i0} + 0.007x_{i1}^2\,; \\
R2 :\ & p_{i2} = 0.5p_{i0} + 0.02 - 0.01x_{i2}^3\,; \\
R3 :\ & P_{i3} = 0.5p_{i0} + 0.1|x_{i1}|\,; \\
R4 :\ & p_{i4} = 0.5p_{i0} + 0.01 + exp(x_{i2})\,; \\
R5 :\ & p_{i5} = 0.5p_{i0} + 0.2 + 0.1\{(sin(x_{i1}) + cos(x_{i2})\}\,; \\
R6 :\ & p_{i6} = \mathbf{1}_{(x_{i1}<0)}(0.4 + 0.2x_{i4}) + \\
& \mathbf{1}_{(x_{i1}\geq 0)}\mathbf{1}_{(x_{i2}<0.75)}\mathbf{1}_{(x_{i3}<6)}\{0.5\mathbf{1}_{(x_{i5}=1)} + 0.65\mathbf{1}_{(x_{i5}=2)} + 0.7\mathbf{1}_{(x_{i5}=3)}\} \\
& + 0.8\mathbf{1}_{(x_{i1}\geq 0)}\mathbf{1}_{(x_{i2}<0.75)}\mathbf{1}_{(x_{i3}\geq 6)} + 0.9\mathbf{1}_{(x_{i1}\geq 0)}\mathbf{1}_{(x_{i2}\geq 0.75)}\,;
\end{aligned}
$$

Figures presented in Appendix 6.1 show the distributions of the simulated values of response probabilities p_{id}, $d = 0, \cdots, 6$. Note that the resulting response rates are approximately 85% for $R0$, 56% for $R1$, 45% for $R2$, 51% for $R3$, 58% for $R4$, 69% for $R5$ and 61% for $R6$. Figures presented in Appendix 6.2 illustrate the possibility of non linear links between the response probabilities and the survey variables in our simulations : Hajek's estimator is expected to outperform the expansion estimator in such situations.

We use a truncation for $\hat{p}_i$ with a 0.02 lower bound for all the methods (with or without HRG). As a measure of bias of an estimator $\hat{t}_{y_{(m)}}$ of the finite population parameter t_y, using machine learning method m for response probabilities estimations, we compute the Monte Carlo percent relative bias

$$
RB_{MC}(\hat{t}_{y_{(m)}}) = \frac{1}{K}\sum_{k=1}^{K}\frac{(\hat{t}_{y_{(m,k)}} - t_y)}{t_y} \times 100, \tag{6}
$$

where $\hat{t}_{y_{(m,k)}}$ denotes the estimator of t_y in the k-th sample obtained with machine learning method m. As a measure of relative efficiency, we compute

$$
RE_{MC}(\hat{t}_{y_{(m)}}) = \frac{MSE_{MC}(\hat{t}_{y_{(m)}})}{MSE_{MC}(\hat{t}_{y_{(HRG\ \ Reglog)}})}, \tag{7}
$$

where $\hat{t}_{y_{(m)}}$ and $\hat{t}_{y_{(HRG\ \ Reglog)}}$ denote respectively the estimator of t_y obtained with method m and the estimator of t_y obtained with Homogenous Response Group applied to logistic regression estimated probabilities, and where

$$
MSE_{MC}(\hat{t}_{y_{(m)}}) = \frac{1}{K}\sum_{k=1}^{K}(\hat{t}_{y_{(m,k)}} - t_y)^2.
$$

Using $RB_{MC}(\hat{t}_{y_{(m)}})$ and $RE_{MC}(\hat{t}_{y_{(m)}})$ as measures of performance leads to a huge amounts of indicators. Indeed, we have to cross 7 response mechanisms, by 10 variables of interest, 30 methods (with and without HRG versions of 15 machine learning methods) and this for 2 types of estimators $\hat{t}_{y_{Exp}}$ and $\hat{t}_{y_{Haj}}$: 42000 performance indicators. We have to sum up all this information. In order to get a global ranking of the 30 methods for $\hat{t}_{y_{Exp}}$ and $\hat{t}_{y_{Haj}}$, we build two kind of global indicators : one to sum up the RB_{MC} tables and one to sum up the RE_{MC} tables of each machine learning method.

4.2 Relative Bias results

Global indicator of relative bias

For each machine learning method, we have a RB_{MC} table containing **70 indicators** (10 rows for the 10 variables of interest and 7 columns for the 7 response mechanisms) that can be **summed up by one indicator : the Frobenius norm of the $\mathbf{RB_{MC}}$ table**. The definition of the Frobenius norm of a matrix T is $\|T\|_F = \sqrt{trace(T^*T)}$ where T^* is the conjugate transpose of T. We want to identify the methods with the lowest relative bias. Thus we look for the methods for which the Frobenius norm of relative bias tables are the smallest. Once we get the global ranking of the methods based on this norm, we can go into more details for the best methods.

Global ranking results

In terms of relative bias results summed up with $\|RB_{MC}\|_F$ (figure 2), the best method is HRG logistic regression for both $\hat{t}_{y_{Exp}}$ and $\hat{t}_{y_{Haj}}$. However, among the methods that could handle missing values in predictors, HRG unpruned CART is good and performs better than unpruned CART (and much better than default pruned CART and than HRG prunned CART). Bagging Ctree (which also could handle missing values in predictors) performs also quite good but better for $\hat{t}_{y_{Haj}}$ than for $\hat{t}_{y_{Exp}}$. As shown in figure 2, the four best methods for $\hat{t}_{y_{Exp}}$ provide lower bias than the four best for $\hat{t}_{y_{Haj}}$. We also can see that applying HRG reduces bias for the very best methods (logistic regression and Unpruned CART) but it is not the case for all the methods (see for instance Bagging Ctree and MultiVariate CTrees). Note that in figure 2, the most extreme values have been removed for a better readability : only the 25 best methods (among 30) are provided.

a. RB_{MC} **: Focus on the three best methods for** $\hat{t}_{y_{Exp}}$

a.1 ***HRG logistic regression (Table 1, Frobenious norm = 22.5)***
Among the 70 scenarios, 30 show unbiased $\hat{t}_{y_{Exp}}$ (bias $< 1\%$) and 9 scenarios exhibit bias above 4%. The best results occur with $R0$ (reference response mechanism i.e. logit link) and $R6$ (decision tree response mechanism). The worse results occur with $R2$ (reference response mechanism + a quadratic term) and $R4$ (reference response mechanism + an exponential term). The highest bias equals -7.7 with $Y7$ (model with quadratic, cubic and interaction terms) and $R5$ (reference response mechanism + sine and cosine terms).

a.2 ***HRG Unpruned CART (Table 2, Frobenious norm = 26.36)***
Among the 70 scenarios, 17 show unbiased $\hat{t}_{y_{Exp}}$ (bias $< 1\%$) and 13 scenarios exhibit bias above 4%. The best results occur with $R0$ (reference response mechanism i.e. logit link) and $R6$ (decision tree response mechanism). The worse results occur with $R2$ (reference response mechanism + a quadratic term) and $R3$ (reference response mechanism + an absolute value term). The highest bias equal -8.49% with $Y10$ and $R2$ (reference response mechanism + a quadratic term) and -7.22% with $Y10$ (model with a cosine term) and $R3$ (reference response mechanism + an absolute value term).

a.3 ***Logistic (Table 3, Frobenious norm = 27.62)***
Among the 70 scenarios, 27 show unbiased $\hat{t}_{y_{Exp}}$ (bias $< 1\%$) and 14 scenarios exhibit bias above 4%. The best results occur with $R0$ (reference response mechanism i.e. logit link) and $R6$ (decision tree response mechanism). The worse results occur with $R2$ (reference response mechanism + a quadratic term) and $R4$ (reference response mechanism + an exponential term). The highest relative bias equals 9.28% with $Y7$ (model quadratic, cubic and interaction terms) and $R4$ (reference response mechanism + an exponential term).

FIGURE 2 – Frobenius norm of the relative bias tables for $\hat{t}_{y_{Exp}}$ and $\hat{t}_{y_{Haj}}$

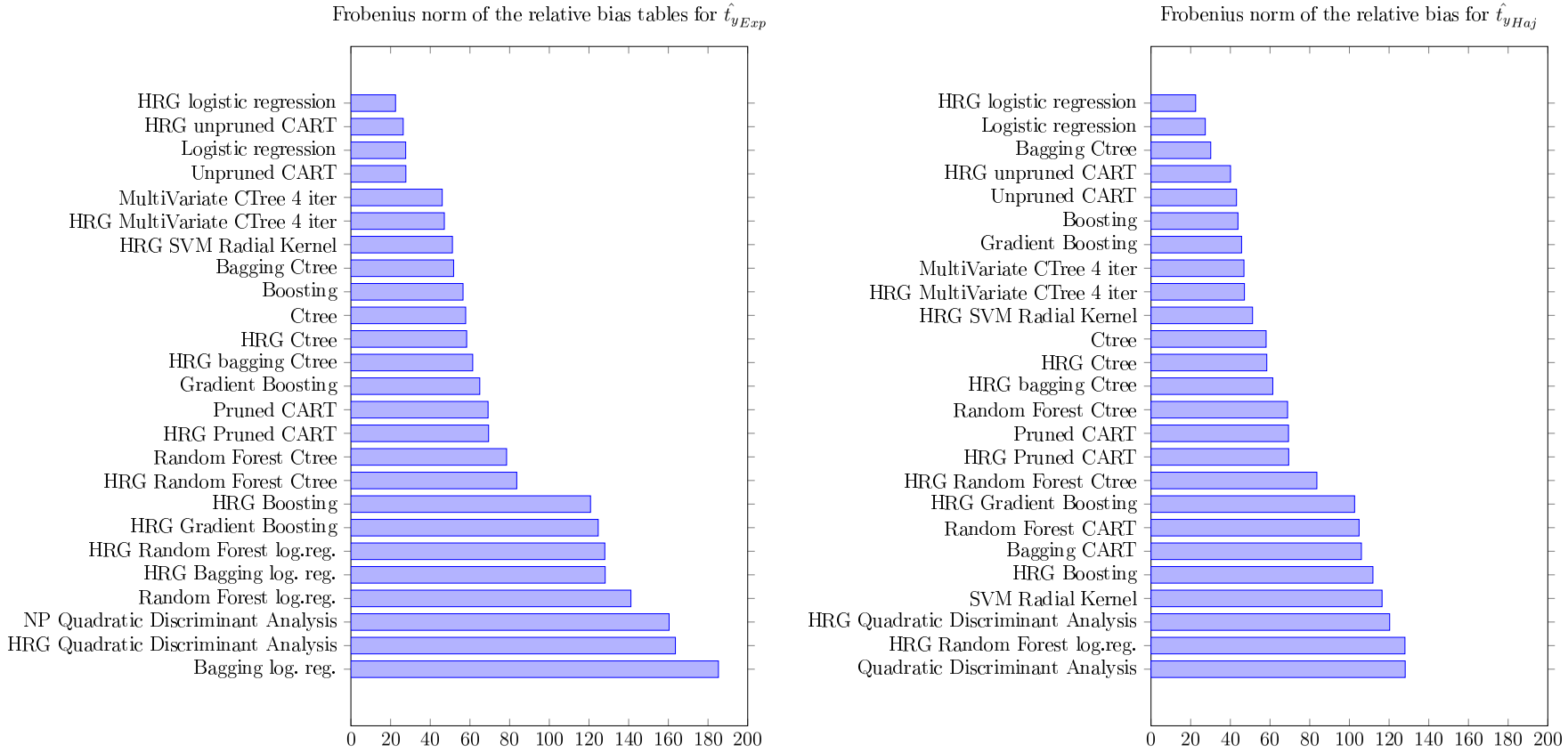


TABLE 1 – Relative bias of $\hat{t}_{y_{Exp}}$ with HRG after logistic regression

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	0.82	3.65	5.30	3.35	4.60	2.49	- 0.33
Y2	0.60	2.68	3.85	2.44	3.33	1.84	- 0.23
Y3	- 0.12	0.19	0.69	0.78	0.70	0.35	- 0.04
Y4	0.35	2.40	3.77	2.77	3.39	1.88	- 0.24
Y5	0.11	2.91	5.28	3.62	4.67	2.35	- 0.80
Y6	0.07	1.71	3.41	0.84	4.33	- 1.46	2.27
Y7	0.85	2.79	5.91	- 0.22	6.99	- 7.68	- 2.32
Y8	1.19	- 1.29	- 1.51	0.63	- 3.06	2.55	- 0.26
Y9	0.72	- 0.86	- 1.80	- 0.23	- 2.57	1.95	- 0.92
Y10	0.15	0.03	0.64	- 4.70	1.21	- 0.99	0.16

TABLE 2 – Relative bias of $\hat{t}_{y_{Exp}}$ with HRG after unpruned CART

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	2.53	2.33	- 2.13	- 0.95	2.61	2.29	- 0.88
Y2	1.86	0.75	- 4	- 2.37	0.58	1.39	- 1.26
Y3	0.36	- 2.81	- 7.56	- 5.07	- 3.66	- 0.23	- 1.94
Y4	1.82	0.91	- 3.70	- 2.06	0.65	1.62	- 1.11
Y5	2.61	3.32	- 1.13	0.09	3.66	2.76	- 1.36
Y6	0.77	- 2.05	- 4.68	- 4.81	- 4.03	- 0.34	- 1.10
Y7	3.41	2.12	0.70	- 2.87	0.47	1.21	- 0.43
Y8	3.78	3.49	- 1.02	- 1.38	5.32	5.18	2.44
Y9	3.14	5.23	- 4.77	- 0.36	3.61	4.37	3.68
Y10	- 0.02	- 3.72	- 8.49	- 7.22	- 4.80	- 0.71	- 2.57

TABLE 3 – Relative bias of $\hat{t}_{y_{Exp}}$ with Logistic regression

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	0.06	3.56	5.25	2.95	4.08	2.61	- 0.71
Y2	0.04	2.55	3.77	2.15	3.07	1.85	- 0.42
Y3	- 0.21	- 0.01	0.71	0.76	1.16	0.15	0.00
Y4	- 0.17	2.32	3.78	2.51	3.29	1.88	- 0.59
Y5	- 0.70	3.06	5.42	3.18	4.18	2.53	- 1.16
Y6	- 0.03	0.51	3.23	0.66	6.80	- 1.86	2.46
Y7	- 0.25	1.69	5.93	- 0.85	9.28	- 8.42	- 5.80
Y8	- 0.11	- 7.28	- 6.80	- 1.14	- 3.54	- 0.38	0.72
Y9	- 0.49	- 6.22	- 6.50	- 1.52	- 2.90	- 0.82	- 0.13
Y10	0.01	0.27	1.29	- 5.13	1.60	- 0.59	- 0.62

b. RB_{MC} **: Focus on the three best methods for** $\hat{t}_{y_{Haj}}$

b.1 ***HRG logistic regression (Table 4, Frobenious norm = 22.5)***
Among the 70 scenarios, 30 show unbiased $\hat{t}_{y_{Exp}}$ (bias $< 1\%$) and 9 scenarios exhibit bias above 4%. The best results occur with $R0$ (reference response mechanism i.e. logit link) and $R6$ (decision tree response mechanism). The worse results occur with $R2$ (reference response mechanism + a quadratic term) and $R4$ (reference response mechanism + an exponential term). The highest relative bias equals -7.68% with $Y7$ (model with quadratic, cubic and interaction terms) and $R5$ (reference response mechanism + sine and cosine terms).

b.2 ***Logistic regression (Table 5, Frobenious norm = 27.36)***

Among the 70 scenarios, 28 show unbiased $\hat{t}_{y_{Exp}}$ (bias $< 1\%$) and 12 scenarios exhibit bias above 4%. The best results occur with $R0$ (reference response mechanism i.e. logit link). The worse results occur with $R2$ (reference response mechanism + a quadratic term). The highest relative bias equals 8.81% with $Y7$ (model with quadratic, cubic and interaction terms) and $R4$ (reference response mechanism + an exponential term).

b.3 ***Bagging Ctree (Table 6, Frobenious norm = 30.11)***
Among the 70 scenarios, 23 show unbiased $\hat{t}_{y_{Exp}}$ (bias $< 1\%$) and 21 scenarios exhibit bias above 4%. The best results occur with $R0$ (reference response mechanism i.e. logit link) and $R6$ (decision tree response mechanism). The worse results occur with $R2$ (reference response mechanism + a quadratic term). The highest relative bias equals -8.62% with $Y10$ (model with a cosine term) and $R2$ (reference response mechanism + a quadratic term).

TABLE 4 – Relative bias of $\hat{t}_{y_{Haj}}$ with HRG after logistic regression

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	0.82	3.65	5.30	3.35	4.60	2.49	-0.33
Y2	0.60	2.68	3.85	2.44	3.33	1.84	-0.23
Y3	-0.12	0.19	0.69	0.78	0.70	0.35	-0.04
Y4	0.35	2.40	3.77	2.77	3.39	1.88	-0.24
Y5	0.11	2.91	5.28	3.62	4.67	2.35	-0.80
Y6	0.07	1.71	3.41	0.84	4.33	-1.46	2.27
Y7	0.85	2.79	5.91	-0.22	6.99	-7.68	-2.32
Y8	1.19	-1.29	-1.51	0.63	-3.06	2.55	-0.26
Y9	0.72	-0.86	-1.80	-0.23	-2.57	1.95	-0.92
Y10	0.15	0.03	0.64	-4.70	1.21	-0.99	0.16

TABLE 5 – Relative bias of $\hat{t}_{y_{Haj}}$ with Logistic regression

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	0.08	3.85	5.33	2.93	3.64	2.85	-0.98
Y2	0.06	2.84	3.85	2.14	2.63	2.09	-0.70
Y3	-0.20	0.27	0.78	0.74	0.73	0.39	-0.28
Y4	-0.15	2.61	3.86	2.50	2.85	2.11	-0.87
Y5	-0.68	3.35	5.50	3.16	3.74	2.77	-1.43
Y6	-0.02	0.78	3.30	0.65	6.34	-1.64	2.18
Y7	-0.23	1.98	6.00	-0.87	8.81	-8.22	-6.06
Y8	-0.09	-7.02	-6.75	-1.16	-3.96	-0.15	0.44
Y9	-0.47	-5.96	-6.44	-1.55	-3.33	-0.59	-0.40
Y10	0.03	0.55	1.36	-5.15	1.17	-0.36	-0.90

TABLE 6 – Relative bias of $\hat{t}_{y_{Haj}}$ with Ctree Bagging

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	1.46	5.26	6.30	5.78	5.44	1.85	0.67
Y2	1.13	3.90	4.54	4.22	3.99	1.36	0.50
Y3	-0.05	0.73	0.81	0.89	0.97	0.59	-0.47
Y4	0.84	3.86	4.45	4.31	4.20	1.75	-0.13
Y5	0.16	5.16	6.39	6.20	5.81	2.03	-0.11
Y6	2.32	3.67	4.29	3.45	4.13	2.09	0.54
Y7	2.81	7.65	9.13	7.08	9.34	3.01	0.01
Y8	2.86	0.67	1.32	4.84	2.22	4.04	2.16
Y9	2.41	1.19	0.83	3.02	2.56	3.71	1.35
Y10	-0.88	-0.02	0.39	-3.02	0.38	0.03	-0.11

4.3 Relative Efficiency results

Global indicator of relative efficiency

In the definition of relative efficiency RE_{MC} (equation 7), we explicitly use the logistic regression combined with HRG as the reference method. It is not the case in the definition of relative bias RB_{MC} (equation 6). That is why we propose a different global indicator of performance, normalized to 1 for the logistic regression combined with HRG.

Let us denote $RE_{MC}(e, m)$ the table computed for :
- e the estimator type of t_y's, $e \in \{\hat{t_{y_{Exp}}}, \hat{t_{y_{Haj}}}\}$,
- m the machine learning method used to estimate response probabilities.

Note that the model m can either be a machine learning used alone to estimate probabilities or a machine learning method associated to the Homogeneous Response Group creation (see section 3.1).

We compute the following normalized indicator (based on the Frobenius norm) :

$$NREF_{(e,m)} = \|RE_{MC}(e, m)/RE_{MC}(e, \text{HRG logistic regression})\|_F/8.3666$$

where $RE_{MC}(e, m)/RE_{MC}(e, \text{HRG logistic regression})$ is a term by term division of $RE_{MC}(e, m)$ by $RE_{MC}(e, \text{HRG logistic regression})$. The denominator 8.3666 is the Frobenius norm of a 10×7 matrix filled with $1's$: it is the Frobenius norm of the table

$RE_{MC}(e, \text{HRG logistic regression})/RE_{MC}(e, \text{HRG logistic regression})$.

Global ranking results

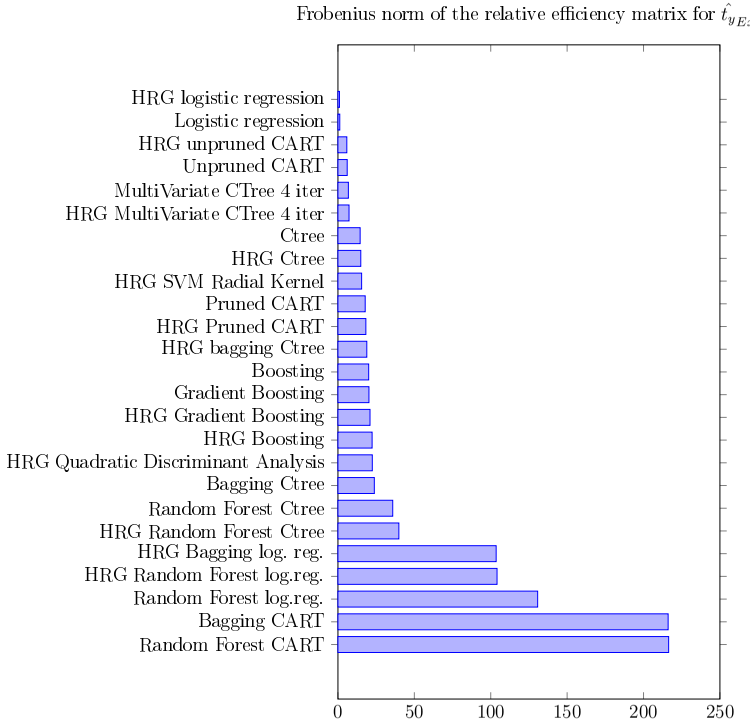
In the Bar plot (Figure 3) the most extreme values of $NREF$ have been removed for a better readability. The best methods are HRG logistic regression, Logistic regression HRG Unpruned CART and Unpruned CART for both $\hat{t}_{y_{Exp}}$ and $\hat{t}_{y_{Haj}}$. However among the methods that could handle missing values in predictors, MultiVariate CTree with four iterations is not far, particularly for $\hat{t}_{y_{Exp}}$.

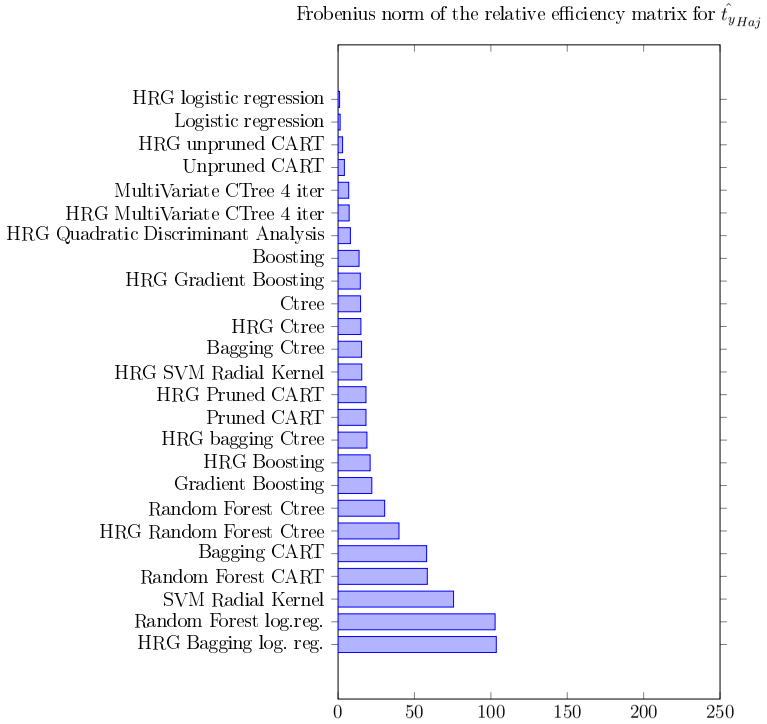
a. RE_{MC} : Focus on the three best methods for $\hat{t}_{y_{Exp}}$

HRG logistic regression is a common used method and appears to the best rank among all the machine learning methods we used. That is why we used it as the reference : data table 7 is used as denominator in RE_{MC} computation for all the other methods. Consequently, it's RE_{MC} table is filled with 1's only which leads to a Frobenius norm equal to 3.87 and a Normalized Frobenius norm equal to 1. Thus we rather provide here the MSE_{MC} table. In the following table, we darkened the worse cases for each variable of interest (the maximal value in each row). It shows for each variable of interest, on which response mechanism HRG logistic regression performs the best (always $R0$ i.e. the reference response mechanism with logit link)) and the worse ($R2$ i.e. the reference response mechanism + a quadratic term for $Y1$ to $Y5$, $R3$ for $Y8$ to $Y10$ for instance).

Let us focus now on the two other best methods in terms of RE_{MC}.

FIGURE 3 – Normalized Frobenius norm of the relative efficiency tables for $\hat{t}_{y_{Exp}}$ and $\hat{t}_{y_{Haj}}$





a.1 ***Logistic regression (Table 8, Normalized Frobenious norm = 1.2)***
Among the 70 scenarios, Logistic regression outperforms HRG Logistic regression in 36 scenarios ($RE_{MC} < 1$)) and is much worse in 3 scenarios with $RE_{MC} > 2$. The relative best outperformances of Logistic regression occur with $R1$ (logit response mechanism with non normal residuals) and $R2$ (reference response mechanism + a quadratic term). The worse underperformances occur with $Y5$ and $R0$ (reference response mechanism i.e. logit link) : $RE_{MC} = 3.41$ which means that the MSE of Logistic regression is more than three times the one of HRG Logistic regression.

a.2 ***HRG Unpruned CART (Table 9, Normalized Frobenius norm = 5.8)***
Among the 70 scenarios, HRG Unpruned CART outperforms HRG Logistic regression in 18 scenarios ($RE_{MC} < 1$) and is much worse in 25 scenarios with a RE_{MC} higher than 2. Relative underperformances occur with $R0$ to $R6$. The highest RE_{MC} equals 29.41 with $Y10$ and $R2$ (reference response mechanism + a quadratic term) and 20.69 with $Y3$ and $R2$.

TABLE 7 – MSE_{MC} for $\hat{t}_{y_{Exp}}$ with HRG logistic regression

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	1.40E+03	2.14E+04	3.86E+04	2.30E+04	3.07E+04	1.12E+04	8.99E+03
Y2	1.42E+03	2.17E+04	3.94E+04	2.30E+04	3.02E+04	1.19E+04	8.98E+03
Y3	6.51E+02	4.14E+03	8.10E+03	6.62E+03	5.71E+03	3.27E+03	2.27E+03
Y4	1.33E+03	2.66E+04	5.33E+04	3.50E+04	4.27E+04	1.67E+04	9.75E+03
Y5	1.10E+03	1.70E+04	4.00E+04	2.59E+04	3.12E+04	1.10E+04	8.06E+03
Y6	2.72E+03	2.01E+04	4.05E+04	1.80E+04	5.16E+04	1.29E+04	2.15E+04
Y7	3.30E+04	9.16E+04	1.48E+05	1.01E+05	1.57E+05	1.79E+05	5.90E+04
Y8	1.96E+17	3.08E+20	3.94E+20	4.38E+20	2.04E+20	3.42E+19	1.79E+20
Y9	1.68E+17	2.83E+20	3.60E+20	4.23E+20	2.00E+20	3.27E+19	1.75E+20
Y10	1.54E+03	5.79E+03	7.11E+03	6.51E+04	9.55E+03	5.22E+03	4.11E+03

TABLE 8 – Relative efficiency for $\hat{t}_{y_{Exp}}$ with logistic regression

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	1.52	0.91	0.97	0.89	0.90	1.03	1.05
Y2	1.43	0.89	0.96	0.90	0.93	0.97	1.02
Y3	1.33	0.86	0.93	1.03	1.57	0.87	1.03
Y4	1.97	0.90	1.00	0.88	0.99	0.97	1.10
Y5	3.41	0.97	1.01	0.88	0.92	1.05	1.03
Y6	1.23	0.63	0.92	0.93	2.33	1.13	1.11
Y7	1.80	0.70	0.98	1.05	1.65	1.19	2.67
Y8	0.01	1.25	0.56	1.03	0.88	0.48	0.97
Y9	0.01	1.28	0.53	1.05	0.88	0.48	0.97
Y10	1.04	0.97	1.72	1.16	1.36	0.79	1.43

TABLE 9 – Relative efficiency for $\hat{t}_{y_{Exp}}$ with HRG after unpruned CART

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	7.02	1.05	0.70	0.95	0.90	1.51	2.27
Y2	7.23	0.85	1.30	1.35	0.72	1.31	2.73
Y3	3.30	8.12	20.69	12.52	8.86	3.17	10.67
Y4	10.16	0.89	1.25	1.16	0.65	1.36	3.16
Y5	9.40	1.70	0.65	0.88	1.15	1.92	2.57
Y6	3.09	1.74	1.80	3.98	1.27	1.60	1.25
Y7	2.20	1.52	1.00	1.06	0.89	0.66	2.11
Y8	0.08	0.09	0.57	1.84	1.89	1.97	0.63
Y9	0.05	0.02	0.53	1.88	1.72	2.04	0.63
Y10	1.56	8.78	29.41	2.37	8.00	1.87	7.87

b. RE_{MC} : Focus on the three best methods for $\hat{t}_{y_{Haj}}$

Here again, HRG logistic regression is used as the reference (denominator in RE_{MC} computation). In the following table, we darkened the worse cases for each variable of interest (the maximal value in each row). It shows that HRG logistic regression performs the best with $R0$ (reference response mechanism) and the worse with $R2$ (reference response mechanism + a quadratic term) for $Y1$ to $Y6$, $R3$ for $Y8$ to $Y10$.

Let us focus now on the two other best methods in terms of RE_{MC}.

b.1 ***Logistic regression (Normalized Frobenius norm = 1.4)***
Among the 70 scenarios, logistic regression outperforms HRG Logistic regression in 31 scenarios ($RE_{MC} < 1$) and is much worse in 5 scenarios with a RE_{MC} higher than 2. The relative best outperformances occur with $R2$ (reference response mechanism + a quadratic term) and the worse underperformances with $R0$ (reference response mechanism). The highest RE_{MC} equals 4.57 with $Y5$ and $R0$ (reference response mechanism).

b.2 ***HRG Unpruned CART (Normalized Frobenius norm = 2.9)***
Among the 70 scenarios, HRG Unpruned CART outperforms HRG Logistic regression in 12 scenarios ($RE_{MC} < 1$) and is much worse in 42 scenarios with a RE_{MC} higher than 2. The relative best outperformances occur with $Y8$ and $Y9$ and worse underperformances occur with $R0$ (reference response mechanism). The highest RE_{MC} equals 10.37 with $Y5$ and $R0$ (reference response mechanism).

TABLE 10 – MSE_{MC} for $\hat{t}_{y_{Haj}}$ with HRG logistic regression

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	1.40E+03	2.15E+04	3.86E+04	2.30E+04	3.08E+04	1.12E+04	9.00E+03
Y2	1.42E+03	2.17E+04	3.95E+04	2.30E+04	3.03E+04	1.19E+04	8.98E+03
Y3	6.52E+02	4.15E+03	8.11E+03	6.62E+03	5.72E+03	3.27E+03	2.28E+03
Y4	1.33E+03	2.66E+04	5.33E+04	3.50E+04	4.27E+04	1.67E+04	9.76E+03
Y5	1.10E+03	1.70E+04	4.00E+04	2.59E+04	3.13E+04	1.11E+04	8.07E+03
Y6	2.73E+03	2.02E+04	4.05E+04	1.80E+04	5.16E+04	1.29E+04	2.15E+04
Y7	3.30E+04	9.17E+04	1.49E+05	1.01E+05	1.58E+05	1.80E+05	5.90E+04
Y8	1.97E+17	3.08E+20	3.95E+20	4.39E+20	2.04E+20	3.43E+19	1.79E+20
Y9	1.68E+17	2.83E+20	3.60E+20	4.24E+20	2.00E+20	3.27E+19	1.75E+20
Y10	1.54E+03	5.80E+03	7.11E+03	6.51E+04	9.56E+03	5.22E+03	4.12E+03

TABLE 11 – Relative efficiency of $\hat{t}_{y_{Haj}}$ with logistic regression

A	R0	R1	R2	R3	R4	R5	R6
Y1	2.30	1.01	0.99	0.90	0.81	1.16	1.14
Y2	2.47	1.02	0.98	0.91	0.80	1.14	1.13
Y3	1.33	0.88	0.95	0.99	1.07	0.92	1.09
Y4	3.57	1.04	1.02	0.88	0.83	1.14	1.27
Y5	4.57	1.08	1.03	0.89	0.83	1.18	1.14
Y6	1.12	0.64	0.94	0.91	2.00	1.00	0.99
Y7	1.89	0.74	0.99	1.07	1.51	1.16	2.81
Y8	0.03	1.27	0.56	1.03	0.88	0.49	0.97
Y9	0.03	1.30	0.53	1.05	0.87	0.49	0.97
Y10	1.30	1.14	1.83	1.17	1.01	0.82	1.90

TABLE 12 – Relative efficiency of $\hat{t}_{y_{Haj}}$ with HRG after unpruned CART

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	7.14	2.82	1.97	2.28	2.66	2.13	2.41
Y2	7.38	2.83	1.90	2.27	2.67	2.04	2.46
Y3	3.24	2.69	2.03	2.03	2.21	2.27	3.74
Y4	10.37	3.48	2.19	2.35	2.75	2.13	3.01
Y5	9.53	4.42	2.40	2.59	3.26	2.65	2.35
Y6	3.12	1.64	1.77	1.56	0.59	1.49	1.15
Y7	2.21	2.46	2.72	1.27	1.37	0.72	2.19
Y8	0.08	0.10	0.67	2.09	2.10	1.97	0.71
Y9	0.05	0.03	0.63	2.12	1.89	2.04	0.70
Y10	1.53	1.43	1.61	0.17	0.91	1.12	1.86

5 Discussion

In this article, we conducted a comprehensive simulation study, aiming at a global ranking of different machine learning methods in totals t_y estimation performance through response probabilities estimation. In our simulation set-up with a census context, the best method in terms of MSE is the logistic regression associated with Homogeneous Response Groups creation. This is true both for the expansion estimator and for the Hajek estimator. One drawback of this method is that it does'nt handle missing data among regressors. Unpruned CART associated with Homogeneous Response Groups creation appear among the methods with good performance and that could handle missing values among regressors, particularly with the expansion estimator. Note that those two first methods turn out to be very robust against changes in lower bound truncation of estimated probabilities. Bagging Ctree (which also could handle missing values among regressors) outperforms Unpruned CART associated to Homogeneous Response Groups creation with the Hajek estimator. However, it seems to require a higher level of truncation than the usual 0.02 value.

In further researches, we would like to study deeper our proposed iterated version of multivariate Ctree whose performances are quite good. For instance, which variables of interest pattern makes the Iterated MultiVariate CTrees work or fail ? Furthermore, this method could maybe prove useful in a context of imputation. Another interesting field would be evaluating the performance of the different machine learning methods with missing data among the regressors. We could also enlarge the set of model aggregation with stacking for instance (Wolpert 1992, Breiman 1996, Nocairi et al. 2016). And lastly, evaluating the methods with different complex sampling designs could bring useful information.

References

Agresti, A. (2013). *Categorical Data Analysis*. New York : Wiley-Interscience.

Bang, H., Robins, J. M. (2005). Doubly Robust Estimation in Missing Data and Causal Inference Models. *Biometrics*, 61, 962–973.

Beaumont, J. F. (2005), Calibrated imputation in surveys under a quasi-model-assisted approach, Journal of the Royal Statistical Society Series B (Statistical Methodology), 67(3), 445–458.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140.

Breiman, L. (1996). Stacked Regression, *Machine Learning*, 24(1), 49-64.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. (1984). *Classification and regression trees*. Wadsworth & Brooks/Cole Advanced Books & Software.

Chang, T., Kott, P. (2008). Using Calibration Weighting to Adjust for Nonresponse under a Plausible Model. *Biometrika*, 95(3), 555-571.

Cortes, C., Vapnik, V. (1995). Support-Vector networks. *Machine Learning*, 20, 273-297.

De'ath, G., (2002). Multivariate Regression Trees : A New Technique for Modeling Species-Environment Relationships. *Ecology*, 83(4), 1105-1117.

De'ath G (2014). *mvpart : Multivariate Partitioning*. R package version 1.6-2, URL http : //CRAN.R-project.org/package=mvpart.

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, 24, 123-140

Cortes, C., Vapnik, V. (1995). Support-vector networks. *Machine Learning*. 20(3), 273-297.

Culp, M., Johnson, K., Michailidis, G. (2006). ada : An R Package for Stochastic Boosting. *Journal of Statistical Software*, 17.

Da Silva, D.N., J.D. Opsomer (2006). A kernel smoothing method of adjusting for unit non-response in sample surveys. The Canadian Journal of Statistics, 34, 563–579.

Da Silva, D.N., J.D. Opsomer (2009). Nonparametric propensity weighting for survey nonresponse through local polynomial regression. Survey Methodology, 35, 165–176

Ekholm, A. and Laaksonen, S. (1991). Weighting via response modeling in the Finnish Household Budget Survey. Journal of Official Statistics 3, 325-337.

Eltinge, J.L., Yansaneh, I.S. (1997). Diagnostics for formation of nonresponse adjustment cells, with an application to income nonresponse in the U.S. consumer expenditure survey. Survey Methodology, 23, 33—40

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and computation*, 121, 256-285.

Freund, Y., Schapire, R.E. (1996). Experiments with a new boosting algorithm. In *Machine Learning* : Proceedings of the thirteenth International Conference, 148-156. Morgan Kaufman, San Francisco.

Freund, Y., Schapire, R.E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1), 119-139

Friedman, J., Hastie, T., Tibshirani, R. (2000). Additive logistic regression : a statistical view of Boosting. *The annals of statistics*, 28(2), 337-407.

Friedman, J. (2001). Greedy function approximation : a gradient boosting machine. *Annals of statistics*, 1189-1232.

Friedman, J. (2002). Stochastic Gradient Boosting, *Computational Statistics and Data Analysis*, 38(4), 367-378.

Giommi, A.(1984). On the estimation of the probability of response in finite population sampling (Italian, Societa Italiana di Statistica, Atti della Riunione Scientifica della Societa Italiana, 32.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer, second edition.

Haziza, D. (2009). Imputation and inference in the presence of missing data. *Handbook of statistics*, 29, 215-246.

Haziza, D., Beaumont, J.F. (2007.) On the construction of imputation classes in surveys. *International Statistical Review*, 75(1),25-43.

Haziza, D., Beaumont, J.F. (2017). Construction of weights in surveys : a review. *Statistical Science*, 32, 206-226.

Haziza, D., Lesage, E. (2016). A discussion of weighting procedures for unit nonresponse. Journal of Official Statistics, 32, 129-145.

Haziza, D. and Rao, J. N. K. (2006). A nonresponse model approach to inference under imputation for missing survey data. *Survey Methodology*, 32(4), 53-64.

Ho, T.K. (1995). Random Decision Forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, QC, 14-16, 278-282.

Hosmer, D.W., Lemeshow, S. (2000). *Applied logistic regression*. Wiley Series in Probability and Mathematical Statistics.

Hothorn, T., Hornik, K., Zeileis, A. (2006). Unbiased Recursive Partitioning : A Conditional Inference Framework. *Journal of Computational and Graphical Statistics*, 15(3), 651-674.

Geoffrey, J. McLachlan (2005). *Discriminant Analysis and Statistical Pattern Recognition*. Wiley.

Karatzoglou, A., Meyer, D., Hornik, K. (2006). Support Vector Machines in R. *Journal of Statistical Software*, 15(9).

Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. Applied Statistics 29 (2), 119–127.

Kim, J. K., Kwon, Y., and Park, M. (2016). Calibrated propensity score method for survey nonresponse in cluster sampling. Biometrika, 103 :461– 473.

Li, C. (2016). *A Gentle Introduction to Gradient Boosting*.
URL : http ://www.ccs.neu.edu/home/vip/teach/MLcourse/4_ boosting/slides/gradient_boosting.pdf.

Little, R.J.A. (1986). Survey Nonresponse Adjustments for Estimates of Means. International Statistical Review, 54, 139—157.

Little R. J. A., Vartivarian S. (2005). Does weighting for nonresponse increase the variance of survey means ? , Survey Methodology, 31, 161–168.

McLachlan (2005). *Discriminant Analysis and Statistical Pattern Recognition*, Wiley.

Nakache, J.P., Confais, J. (2003). *Statistique explicative appliquée*, Technip, 206-211.

Niculescu-Mizil, A., Caruana, R. (2005). Obtaining Calibrated Probabilities from Boosting. *Uncertainty in Artificial Intelligence*.

Niculescu-Mizil, A., Caruana, R. (2005). Predicting Good Probabilities with Supervised Learning. *ICML*.

Phipps, P., Toth, D. (2012). Analyzing establishment nonresponse using an interpretable regression tree model with linked administrative data. *Annals of Applied Statistics*, 6(2), 772-794.

Nocairi, H., Gomes, C., Thomas, M., Saporta, G. (2016) Improving Stacking Methodology for Combining Classifiers ; Applications to Cosmetic Industry. *Electronic Journal of Applied Statistical Analysis*, 9(2), 340-361.

Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33, 1065–1076.

Platt, J. (2000). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans (Eds.), *Advances in large margin classifiers*. Cambridge : MIT Press.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27, 832–837.

Rubin, D.B. (1976). Inference and Missing Data. *Biometrika*, 63, 581-590.

Särndal, C. E. and Swensson, B. (1987). A general view of estimation for two-phases of selection with applications to two-phase sampling and non-response. International Statist. Review 55, 279- 294.

Särndal, C.-E., Swensson, B. and Wretman, J. (1992). *Model Assisted Survey Sampling*. Springer-Verlag.

Schapire, R.E. (1990). The Strength of Weak Learnability. *Machine Learning*, Boston, MA : Kluwer Academic Publishers, 5 (2), 197-227.

Schapire, R.E., Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3), 297-336.

Simonoff, J.S. (1996). *Smoothing Methods in Statistics*. Springer.

Schölkopf B., Smola A. (2002). *Learning with Kernels*. MIT Press.

Strasser, H., Weber, C. (1999). On the Asymptotic Theory of Permutation Statistics. *Mathematical Methods of Statistics*, 8, 220–250.

Skinner C. J., D'arrigo (2011). Inverse probability weighting for clustered nonresponse, *Biometrika*, 98, 953–966.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.

Wolpert, D. (1992). Stacked Generalization. *Neural Networks*, 5, 41-259

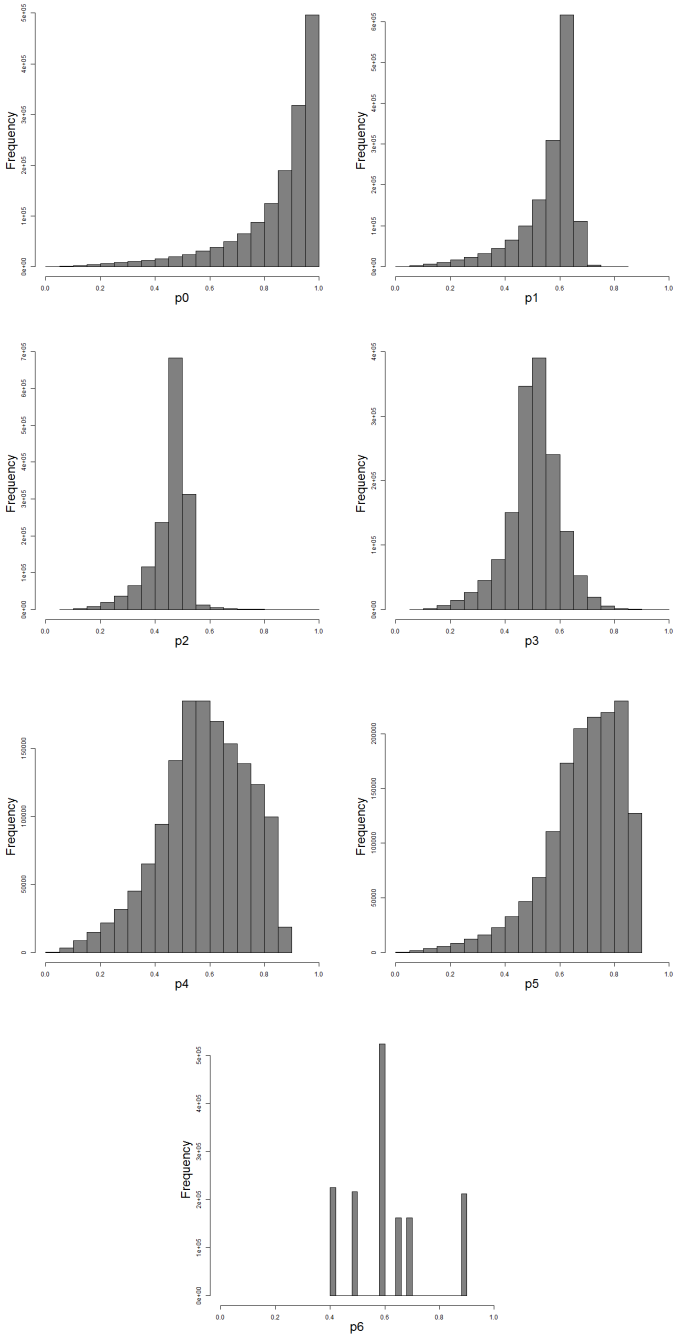
Zadrozny, B., Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. *ICML*, 1, 609-616.

Zhou, Z.-H., (2012). *Ensemble Methods : Foundations and Algorithms*. Chapman & Hall/CRC.

6 Appendix

6.1 Distributions of the generated response probabilities

Distributions on the $K \times N = 1000 \times 1500$ units for the seven response mechanisms



6.2 Plots between response probabilities ($p0$ to $p6$) and variables of interest ($Y1$ to $Y10$)

FIGURE 4 – Scatter plots of $p0$ and variables of interest

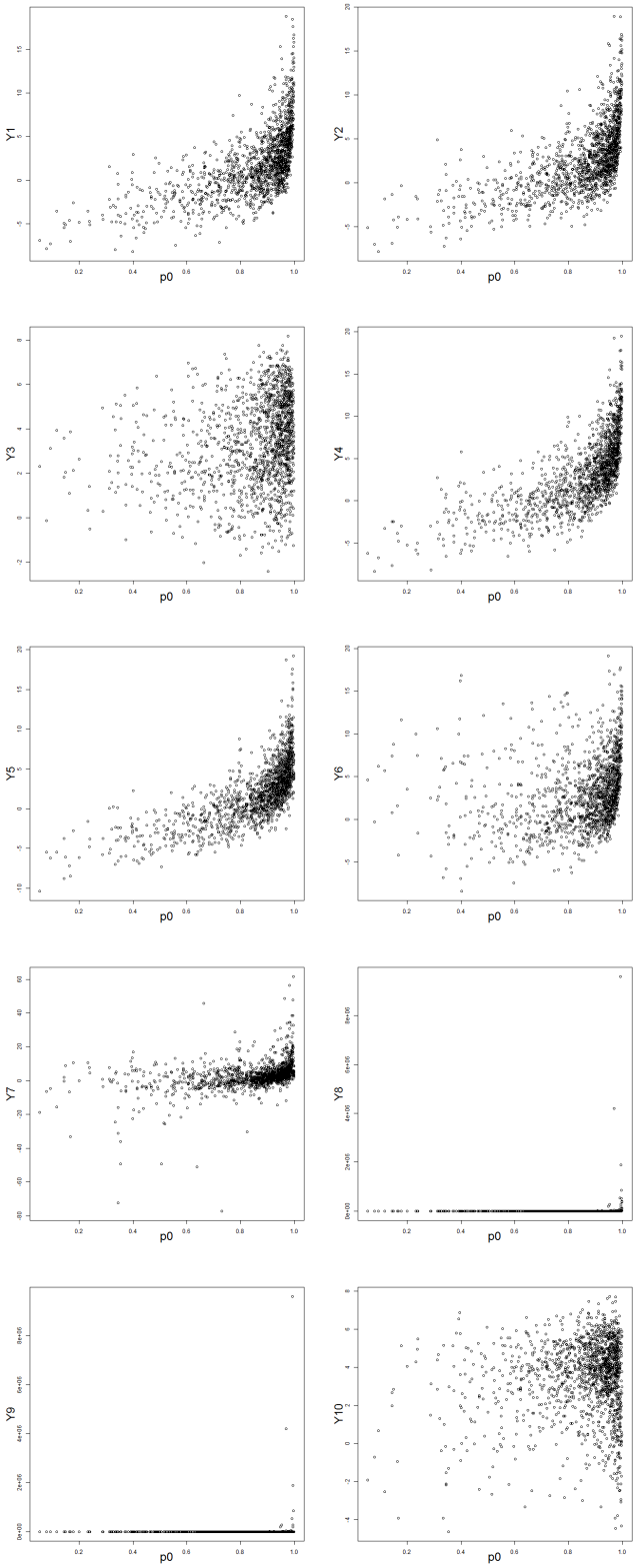


FIGURE 5 – Scatter plots of $p1$ and variables of interest

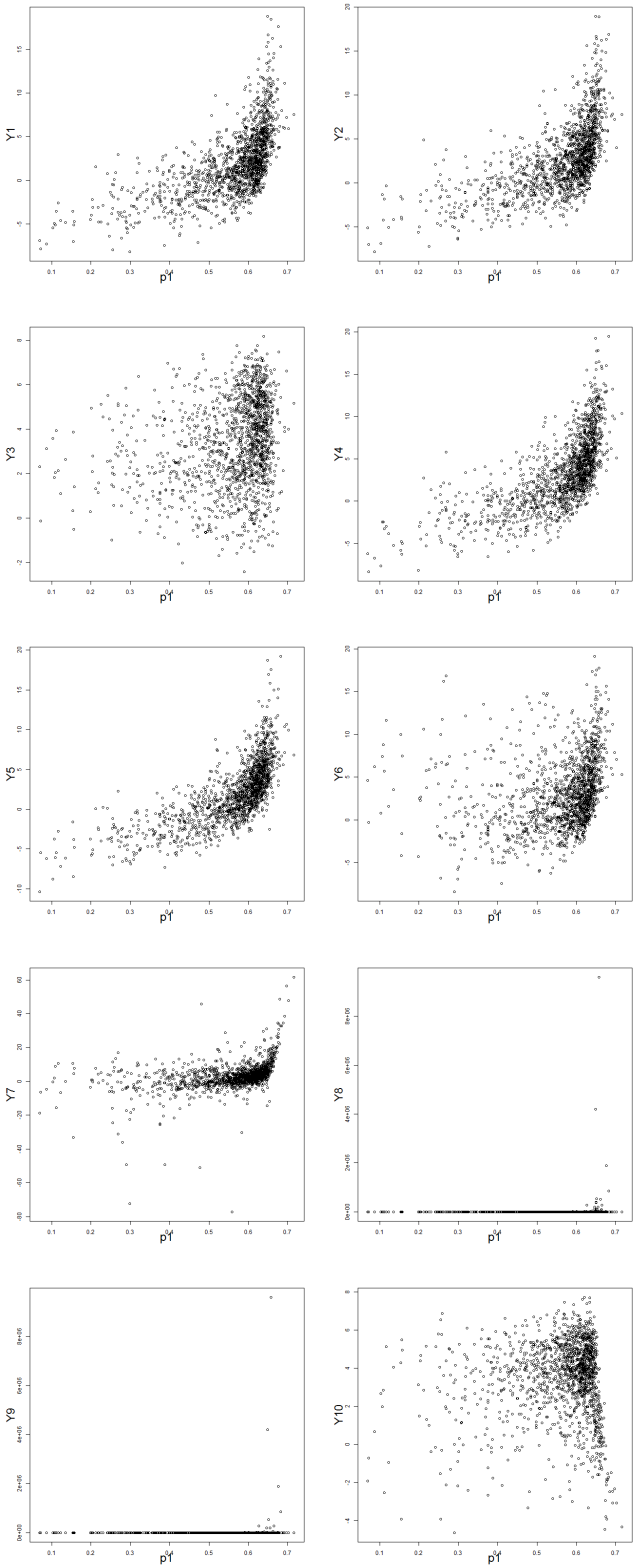


FIGURE 6 – Scatter plots of $p2$ and variables of interest

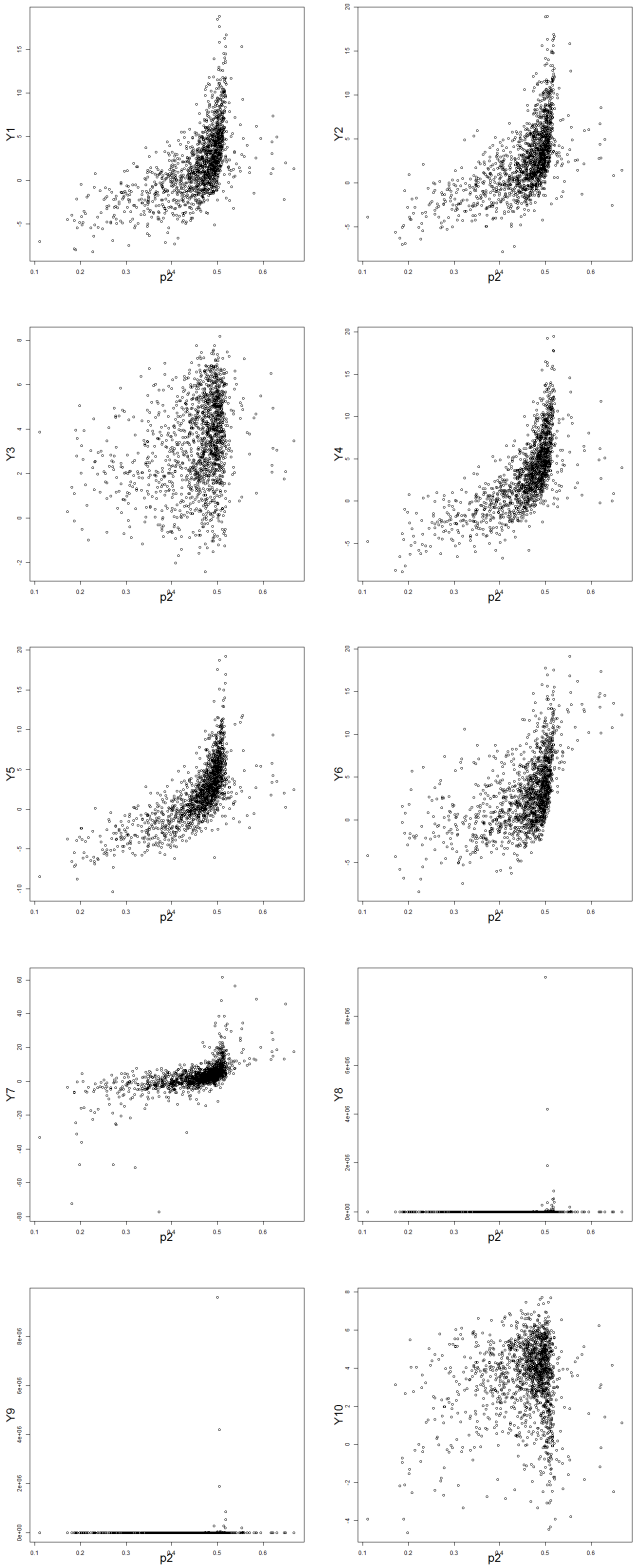


FIGURE 7 – Scatter plots of $p3$ and variables of interest

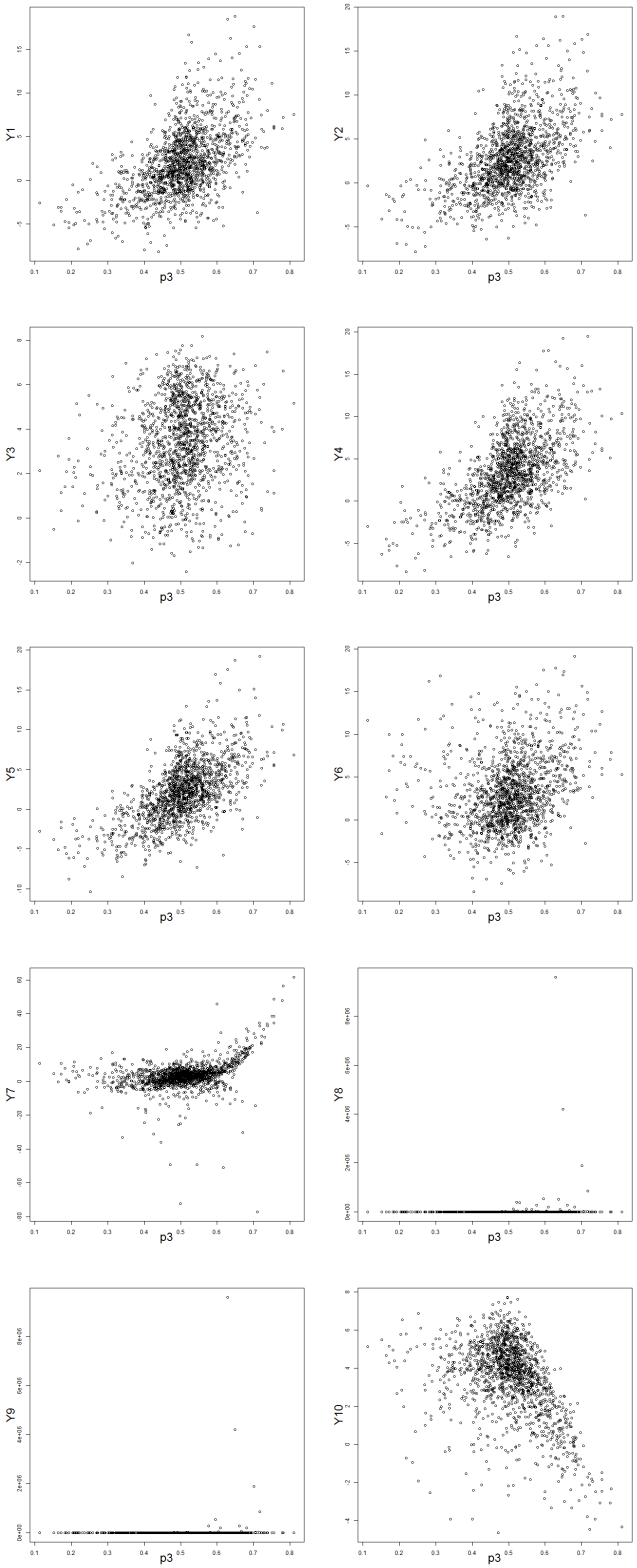


FIGURE 8 – Scatter plots of $p4$ and variables of interest

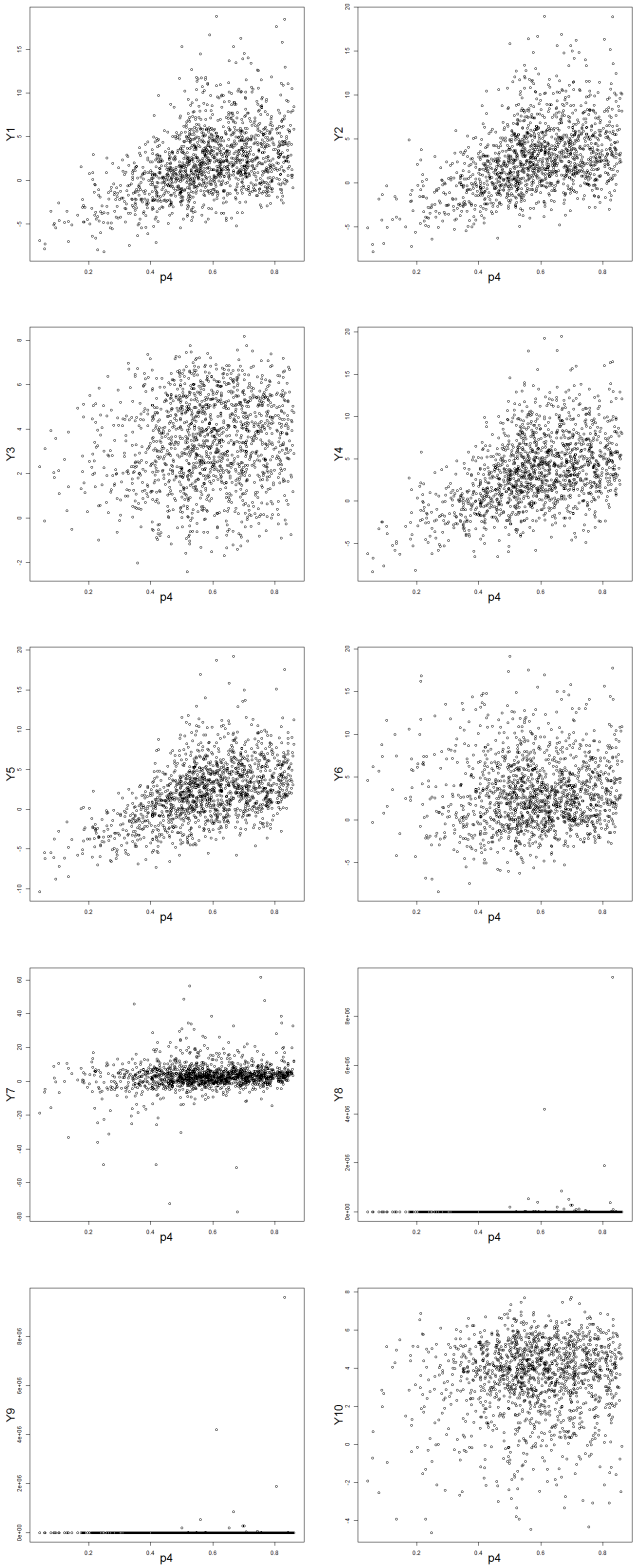


FIGURE 9 – Scatter plots of $p5$ and variables of interest

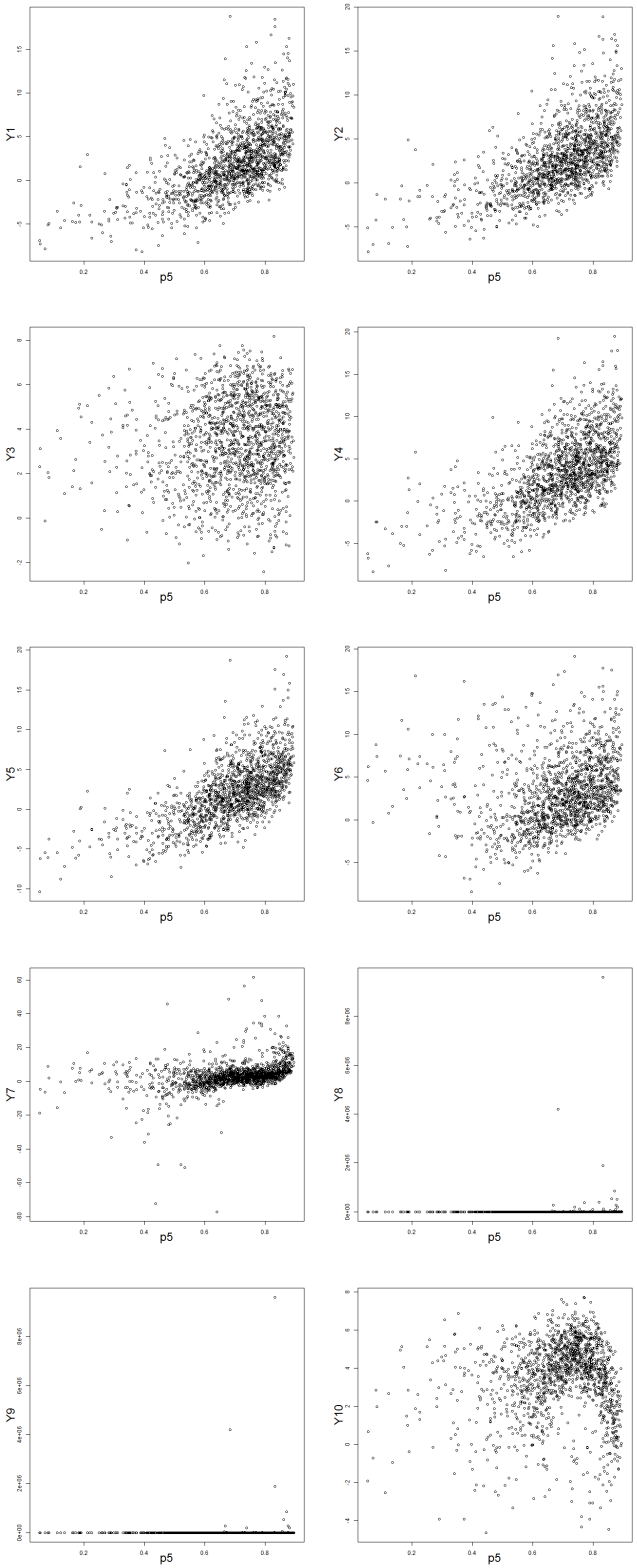
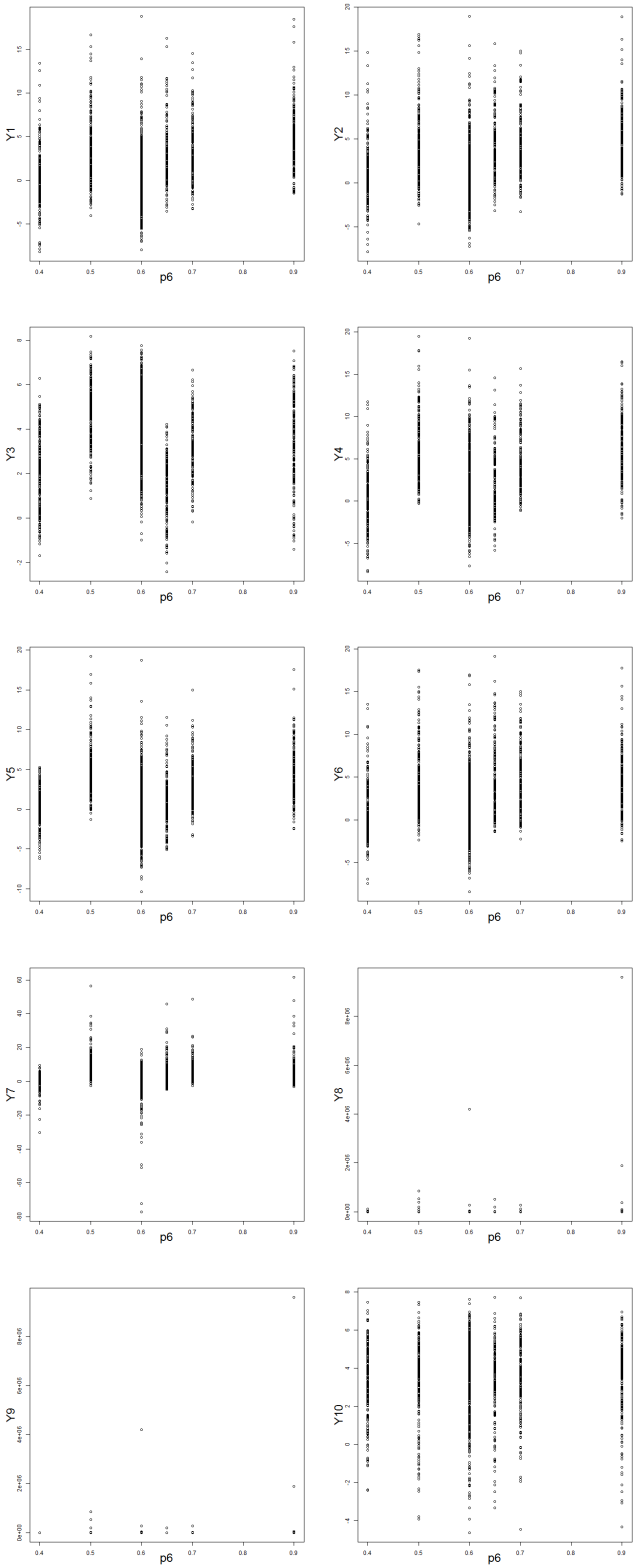


FIGURE 10 – Scatter plots of $p6$ and variables of interest



6.3 Impact of estimated probabilities' truncation

In order to avoid too small values for $\hat{p}_i$, the common practice is to implement truncation with a lower bound t for $\hat{p}_i$'s. A usually implemented lower bound is $t = 0.02$. We want to check how much the choice of a different value in t could change the final performance in terms of MSE for $\hat{t}_y$ built on different machine learning methods. Let us denote $TMSE_{(e,m,t)}$ an MSE table computed for :
- e the estimator type of t_y's, $e \in \{\hat{t_{y_{Exp}}}, \hat{t_{y_{Haj}}}\}$,
- m the machine learning method used to estimate response probabilities,
- t the lower bound used for truncation.

Note that the model m can either be a machine learning used alone to estimate probabilities or a machine learning method associated to the Homogeneous Response Group creation (see section 3.1).

In our simulation study (see section 4), for each combination $e \times m \times t$, we have 70 indicators of MSE for $\hat{t}_y$ (10 variables of interest $\times$ 7 response mechanisms) - see for instance table 13. Thus we need a global indicator to sum up the overall modification of the 70 MSE's induced by a change in t. The Frobenius norm $\|TMSE_{(e,m,t)}\|_F = \sqrt{trace(TMSE^*_{(e,m,t)} TMSE_{(e,m,t)})}$ of the $TMSE_{(e,m,t)}$'s could provide this global measure of performance, and help evaluating the impact of a change in t. Indeed, the lower the MSE's are, the better the combination $e \times m \times t$ is. Thus, given e and m, the best value for t is the one that provides the lowest $\|TMSE_{(e,m,t)}\|_F$.

However, for an easier analysis of the results, we rather compute the following normalized indicator (still based on the computation of a Frobenius norm) :

$$NF_{(e,m,t)} = \|TMSE_{(e,m,t)}/TMSE_{(e,m,t=0.02)}\|_F/8.3666$$

where $TMSE_{(e,m,t)}/TMSE_{(e,m,t=0.02)}$ is a term by term division of $TMSE_{(e,m,t)}$

by $TMSE_{(e,m,t=0.02)}$. The reference value for t is 0.02. The denominator 8.3666 is the Frobenius norm of a 10×7 matrix filled with $1's$: it is the NF value in case of TMSE's global stability when t=0.02 is replaced by an other value of t.

TABLE 13 – $TMSE_{(\hat{t_{y_{Haj}}}, \text{ HRG after logistic regression, } 0.02)}$ for the 10 Variables of interest and the 7 response mechanisms

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	1.4021e+03	2.1458e+04	3.8609e+04	2.2985e+04	3.0776e+04	1.1202e+04	8.9972e+03
Y2	1.4213e+03	2.1680e+04	3.9453e+04	2.3009e+04	3.0271e+04	1.1876e+04	8.9842e+03
Y3	6.5187e+02	4.1476e+03	8.1098e+03	6.6249e+03	5.7192e+03	3.2749e+03	2.2757e+03
Y4	1.3316e+03	2.6639e+04	5.3325e+04	3.5028e+04	4.2719e+04	1.6692e+04	9.7642e+03
Y5	1.0984e+03	1.6990e+04	4.0012e+04	2.5905e+04	3.1258e+04	1.1059e+04	8.0650e+03
Y6	2.7269e+03	2.0168e+04	4.0525e+04	1.8004e+04	5.1633e+04	1.2902e+04	2.1496e+04
Y7	3.3016e+04	9.1681e+04	1.4850e+05	1.0149e+05	1.5752e+05	1.7955e+05	5.9044e+04
Y8	1.9665e+17	3.0832e+20	3.9469e+20	4.3861e+20	2.0440e+20	3.4252e+19	1.7940e+20
Y9	1.6805e+17	2.8333e+20	3.6039e+20	4.2388e+20	2.0022e+20	3.2704e+19	1.7510e+20
Y10	1.5409e+03	5.7964e+03	7.1124e+03	6.5140e+04	9.5585e+03	5.2240e+03	4.1191e+03

For instance in table 14, we can examine in detail the 70 ratios of

$$TMSE_{(\hat{t_{y_{Haj}}},\text{ HRG after logistic regression, 0.06})}/TMSE_{(\hat{t_{y_{Haj}}},\text{ HRG after logistic regression, 0.02})}$$

In this example, a change in truncation bound from t=0.02 to 0.06 has very little impacts (only 3 cases in bold font where the ratios are slightly different from 1). The corresponding indicator $NF_{(\hat{t_{y_{Haj}}},\text{HRG after logistic regression, 0.06})}$ is 1 (see table 16).

TABLE 14 – Ratios of TMSE with truncation 0.06 / TMSE with truncation 0.02
$TMSE_{(\hat{t_{y_{Haj}}},\text{ HRG after logistic regression, 0.06})}/TMSE_{(\hat{t_{y_{Haj}}},\text{ HRG after logistic regression, 0.02})}$

Variable	R0	R1	R2	R3	R4	R5	R6
Y1	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Y2	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Y3	1.00	1.00	1.00	1.00	***0.99***	1.00	1.00
Y4	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Y5	1.00	***1.01***	1.00	1.00	1.00	1.00	1.00
Y6	1.00	1.00	1.00	1.00	***0.99***	1.00	1.00
Y7	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Y8	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Y9	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Y10	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Let us focus on two of the best methods in terms of MSE's (see section 4.3). NF indicators table 15 for $\hat{t_{y_{Exp}}}$ and table 16 for $\hat{t_{y_{Haj}}}$, show that HRG after logistic regression is robust in terms of MSE : we can see that NF indicators are always equal to 1, with m = HRG after logistic regression. HRG after Unpruned CART is quite robust but exhibits better global performance in terms of MSE with $t = 0.06$ both for $\hat{t_{y_{Exp}}}$ and for $\hat{t_{y_{Haj}}}$.

TABLE 15 – NF indicator for $\hat{t}_{y_{Exp}}$ with different lower bounds truncation of $\hat{p}_i$

Method	**Lower bound 0.06**	**Lower bound 0.08**	**Lower bound 0.10**	**Lower bound 0.14**
Logistic regression	0.99	0.98	0.97	0.97
Logistic regression Bagging	1.00	1.00	1.00	1.00
Logistic Random Forest	1.00	1.00	1.00	1.00
Quadratic nonparametric discriminant analysis	1.00	1.00	1.00	1.00
Default pruned CART	1.00	1.00	1.00	1.00
Unpruned CART	1.00	1.01	1.05	1.05
CART Bagging	1.00	1.00	1.00	1.00
CART Random Forest	1.00	1.00	1.00	1.00
CART Boosting	1.02	1.11	1.27	1.27
CART Gradient Boosting	1.00	1.00	1.00	1.00
Ctree	1.00	1.00	1.00	1.00
Ctree Bagging	0.80	0.78	0.76	0.76
Ctree Random Forest	0.93	0.92	0.91	0.91
MultiVariate CTrees	1.00	1.00	1.00	1.00
Radial Kernel SVM	1.00	1.00	1.00	1.00
HRG after Logistic regression	***1.00***	***1.00***	***1.00***	***1.00***
HRG after Logistic regression Bagging	1.00	1.00	1.00	1.00
HRG after Logistic Random forest	1.00	1.00	1.00	1.00
HRG after Quadratic nonparametric HRG after discriminant analysis	1.04	1.09	1.16	1.16
HRG after Default pruned CART	1.00	1.00	1.00	1.00
HRG after Unpruned CART	***0.94***	***0.96***	***1.00***	***1.00***
HRG after CART Bagging	1.00	1.00	1.00	1.00
HRG after CART Random Forest	1.00	1.00	1.00	1.00
HRG after CART Boosting	1.02	1.11	1.27	1.27
HRG after CART Gradient Boosting	1.01	1.08	1.17	1.17
HRG after Ctree	1.00	1.00	1.00	1.00
HRG after Ctree Bagging	1.00	1.00	1.00	1.00
HRG after Ctree random Forest	1.00	1.00	1.00	1.00
HRG after MultiVariate CTrees	1.00	1.00	1.00	1.00
HRG after SVM	1.00	1.00	1.00	1.00

TABLE 16 – NF indicator for $\hat{t}_{y_{Haj}}$ with different lower bounds truncation of $\hat{p}_i$

Method	**Lower bound 0.06**	**Lower bound 0.08**	**Lower bound 0.10**	**Lower bound 0.14**
Logistic regression	0.98	0.97	0.96	0.97
Logistic regression Bagging	1.00	1.00	1.00	1.00
Logistic Random Forest	1.00	1.00	1.00	1.00
Quadratic nonparametric discriminant analysis	1.00	1.00	1.00	1.00
Default pruned CART	1.00	1.00	1.00	1.00
Unpruned CART	1.00	1.01	1.03	1.01
CART Bagging	1.00	1.00	1.00	1.00
CART Random Forest	1.00	1.00	1.00	1.00
CART Boosting	1.00	1.00	1.00	1.00
CART Gradient Boosting	1.00	1.00	1.00	1.00
Ctree	1.00	1.00	1.00	1.00
Ctree Bagging	0.79	0.78	0.77	0.78
Ctree Random Forest	0.95	0.95	0.94	0.95
MultiVariate CTrees	1.00	1.00	1.01	1.00
Radial Kernel SVM	1.00	1.00	1.00	1.00
HRG after Logistic regression	***1.00***	***1.00***	***1.00***	***1.00***
HRG after Logistic regression Bagging	1.00	1.00	1.00	1.00
HRG after Logistic Random forest	1.00	1.00	1.00	1.00
HRG after Quadratic nonparametric HRG after discriminant analysis	0.85	0.83	0.84	0.83
HRG after Default pruned CART	1.00	1.00	1.00	1.00
HRG after Unpruned CART	***0.95***	***0.97***	***0.99***	***0.97***
HRG after CART Bagging	1.00	1.00	1.00	1.00
HRG after CART Random Forest	1.00	1.00	1.00	1.00
HRG after CART Boosting	1.39	5.03	11.21	5.03
HRG after CART Gradient Boosting	0.74	0.69	0.67	0.69
HRG after Ctree	1.00	1.00	1.00	1.00
HRG after Ctree Bagging	1.00	1.00	1.00	1.00
HRG after Ctree random Forest	1.00	1.00	1.00	1.00
HRG after MultiVariate CTrees	1.00	1.00	1.00	1.00
HRG after SVM	1.00	1.00	1.01	1.00